# META-HEURÍSTICA ILS PARA O PROBLEMA DE ROTEAMENTO NO PLANEJAMENTO DO INVENTÁRIO FLORESTAL

**Antonio Almeida de Barros Junior**[1]
e-mail: antonioabj@gmail.com

**Gilson Fernandes da Silva**[2]
e-mail: fernandes5012@gmail.com

**Geraldo Regis Mauri**[1]
e-mail: mauri@cca.ufes.br

**Helio Garcia Leite**[3]
e-mail: hgleite@gmail.com

**Carlos Pedro Boechat Soares**[3]
e-mail: csoares@ufv.br

**Adriano Ribeiro de Mendonça**[2]
e-mail: ribeiroflorestal@yahoo.com.br

[1]Departamento de Computação - Universidade Federal do Espírito Santo - UFES
Alto Universitário, s/nº - Cx Postal 16, Guararema – CEP: 29500-000 - Alegre-ES

[2]Departamento de Ciências Florestais e da Madeira - Universidade Federal do Espírito Santo - UFES
Av. Gov. Lindemberg, nº 316, Centro – CEP: 29550-000 – Jerônimo Monteiro-ES

[3]Departamento de Engenharia Florestal - Universidade Federal de Viçosa – UFV
Avenida Purdue, s/nº, Campus Universitário – CEP: 36570-900 – Viçosa-MG

## RESUMO

O presente trabalho apresenta uma nova abordagem para representar e obter soluções para o Problema de Roteamento de Veículos no planejamento de inventários florestais (PRV-IF). Para tal, foi proposto um modelo adaptado do Problema de Roteamento Periódico de Veículos com Janela de Tempo (PRPVJT) para o problema. Os métodos de solução utilizados foram o método exato *brach-and-cut* por meio do solver CPLEX e a meta-heurística *Iterated Local Search* (ILS). Com a finalidade de analisar o desempenho dos métodos, foram criadas instâncias para o PRV-IF de diferentes tamanhos. Os resultados obtidos demonstram que o método ILS foi satisfatório na solução do problema, especialmente nas maiores instâncias.

**PALAVRAS CHAVE. Inventário Florestal, Problema de Roteamento de Veículos, ILS.**

**Área principal: AG&MA - PO na Agricultura e Meio Ambiente.**

## ABSTRACT

This work presents a new approach to represent and obtain solutions for the Vehicle Routing Problem in the planning of forest inventories (PRV-IF). An adapted model of the Periodic Vehicle Routing Problem with Time Windows (PRPVJT) was proposed for the problem. The exact *brach-and-cut* method using the CPLEX solver and *the Iterated Local Search* (ILS) metaheuristic were used as solution methods. The performance of the applied methods was analyzed by creating instances for the PRV-IF with different sizes. The results show that the method ILS was satisfactory in solving the problem, especially for the larger instances.

**KEY WORDS. Forest Inventory, Vehicle Routing Problem, ILS.**

**Main area: AG&MA - OR in Agriculture and Environment.**


https://proceedings.science/p/85471?lang=pt-br

## 1. Introdução

Dentre as atividades que fazem parte do planejamento tático e operacional de uma empresa florestal, o inventário se destaca por ser uma importante ferramenta para o manejo e o planejamento da produção, o qual pode ser realizado sob diferentes níveis de detalhamento e em diferentes pontos no tempo (SOARES et al., 2011). Equívocos nesse tipo de planejamento podem levar a grandes perdas. Neste sentido, essas empresas realizam seus inventários periodicamente para, entre outras coisas, construir modelos de crescimento e produção, realizar pesquisas e planejar a exploração.

De acordo com Campos e Leite (2017), uma parte significativa do custo total de um inventário está relacionada à coleta de dados. A importância e os custos que envolvem a atividade de inventário florestal faz com que sua realização seja previamente planejada. Em empresas florestais de grande porte, existe uma dificuldade em realizar o planejamento desta atividade, dada a quantidade de parcelas instaladas em seus povoamentos e o período (janela de tempo) em que a atividade deve ser realizada. Para isso, realiza-se um planejamento anual, no qual são definidos quais povoamentos florestais (talhões) serão inventariados, e em quais períodos do ano.

Entre as dificuldades encontradas na realização do planejamento do inventário florestal estão: a limitação do número de equipes, o tempo de medição de cada parcela, o tempo de deslocamento para as unidades amostrais e a distância percorrida. Uma alternativa para auxiliar a redução de custos desta atividade, consiste em otimizar rotas e, consequentemente, o tempo de deslocamento das equipes entre os povoamentos florestais (MENEGUZZI, 2011). Para melhor representar o Planejamento de Inventários Florestais, os termos “local” e “veículo” são substituídos neste trabalho por “talhão” e “equipe”, respectivamente.

Na Figura 1, é apresentado um esquema relacionando as principais variáveis (ponto de partida, ponto de chegada, equipe, talhão e parcela) do problema de roteamento de equipes no planejamento de inventários florestais. No esquema ilustrado (Figura 1), existe um ponto de partida e chegada das equipes, representado pela fábrica. Cada equipe, representada por um veículo, deve sair do ponto de origem, percorrer os talhões previstos em sua programação e retornar ao ponto de destino no final do dia.

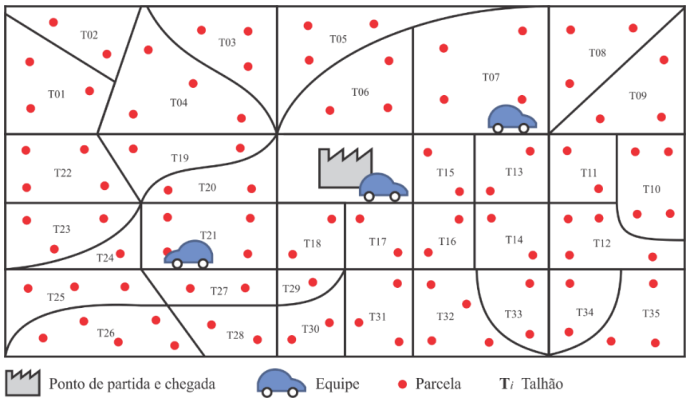



FIGURA 1 – Representação das variáveis envolvidas no PRV-IF.

Na programação das atividades, deve-se considerar um horizonte $H$ de $n$ dias. Para cada dia, define-se uma sequência (rota) de talhões ($T_i$) que deve ser inventariado por uma equipe $k$. O tempo de cada rota (Tempo Total de Serviço - TTS) deve ser inferior à Janela de Tempo Diária (JTD), considerada nesta pesquisa como 8 horas de trabalho diário, ou seja, 480 minutos. Desta forma, o Tempo Total de Serviço (TTS) de uma rota corresponde à soma do Tempo Total de Medição (TTM) com o Tempo Total de Deslocamento (TTD). Sendo TTD a soma dos deslocamentos entre $i$ e $j$ ($td_{ij}$) e TTM a soma do tempo de medição de cada parcela ($t_{si}$). A Figura 2 ilustra a representação destas variáveis.

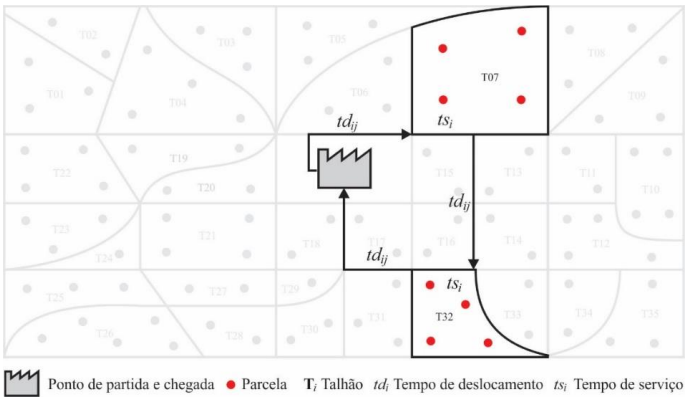



FIGURA 2 – Representação de uma rota diária estabelecida para uma equipe.

A otimização de rotas ou roteamento de veículos é um caso particular de programação linear (PL). Para Arenales et al. (2015), o Problema de Roteamento de Veículos (PRV) exerce uma função preponderante na área de gerenciamento da distribuição e logística. O PRV pode ser aplicado em situações práticas que ocorrem em indústrias, no comércio, no setor de serviços, na segurança, na saúde pública e no lazer.

Goldbarg e Luna (2005) destacam que os problemas de roteamento de veículos estão entre os mais complexos da área de otimização combinatória. Em problemas de grande porte, onde há um número significativo de variáveis, nem sempre é possível a solução por métodos exatos. Dada a insuficiência da resposta dos métodos exatos e a alta complexidade dos problemas de roteamento de veículos faz dos procedimentos aproximativos (heurísticos) uma considerável alternativa quando é preciso encontrar soluções boas em um tempo limitado (GOLDBARG et al., 2015; VIEIRA, 2013).

Face ao exposto, esta pesquisa tem por objetivo apresentar uma metodologia capaz de resolver um Problema de Roteamento de Veículos no Planejamento de Inventários Florestais (PRV-IF), de forma que o custo total da atividade seja minimizado. Para isso, são apresentados resultados obtidos por uso de um método adaptado da meta-heurística clássica *Iterated Local Search* (ILS). O desempenho do método proposto é comparado diretamente com resultados obtidos pelo software CPLEX.

O restante do trabalho é organizado da seguinte forma: a Seção 2 apresenta uma breve revisão da literatura; na Seção 3 é apresentado o modelo matemático e na Seção 4 é apresentada em detalhes a metodologia adotada para a utilização da ILS; os resultados computacionais são apresentados na Seção 5, e as conclusões são descritas na Seção 6.

## 2. Revisão da Literatura

A seguir são descritos trabalhos relacionados ao planejamento de inventários florestais. Os trabalhos de Marcatti (2013) e Haddad (2015) contêm características restritas ao problema abordado nesta pesquisa. Em contrapartida, o trabalho proposto por Meneguzzi (2011) tem uma ampla relação com o problema.

O objetivo do estudo de Marcatti (2013) foi desenvolver metodologias para otimização do acesso e caminhamento às parcelas de inventário florestal, bem como implementá-las em um ambiente com interface gráfica integradas a um sistema de informação geográficas. Para isso, foi proposto um método em duas etapas. A primeira etapa consiste na identificação da entrada e do caminho até a primeira parcela. Na segunda etapa, realiza-se a otimização do caminhamento entre parcelas. O problema foi resolvido via formulação de programação linear inteira e otimizado pelo algoritmo *cutting-plane*. Os resultados obtidos foram comparados com os resultados obtidos pelo método aproximativo do vizinho mais próximo, na tentativa de simular as atividades realizadas intuitivamente pelo executor em campo. Quanto a metodologia aplicada, os resultados mostraram a capacidade de otimização do modelo, visto que a solução exata se mostrou em média 17% superior à solução aproximada.

O trabalho de Haddad (2015) foi realizado com o objetivo de avaliar a viabilidade da aplicação de um modelo de roteamento via Programação Linear Inteira (PLI), visando melhorar o planejamento do inventário florestal urbano da cidade de Belo Horizonte de forma que minimizasse as distâncias percorridas pelas equipes de inventário. Além do método exato, nesse trabalho foi proposta a meta-heurística *Simulated Annealing* para resolução do roteamento, na tentativa de encontrar boas soluções em um tempo inferior ao método exato. Os resultados obtidos por meio do modelo de PLI proporcionou uma melhora de 6,36% em relação ao método tradicional realizado de forma empírica. A meta-heurística *Simulated Annealing* obteve um bom desempenho, já que seu tempo de processamento foi 7.200 vezes menor para obtenção de uma solução viável a 1,87% do valor encontrado pela PLI.

O modelo de Roteamento de Veículos Aplicado ao Planejamento do Inventário Florestal proposto por Meneguzzi (2011) é baseado no modelo de Toth e Vigo (2002) utilizado no Problema de Roteamento de Veículos com Janela de Tempo. O objetivo dessa pesquisa foi de propor um modelo capaz de programar as atividades de inventário para um horizonte de um ano. Para tal, foi selecionado a programação do ano de 2009, que apresentava um total de 1288 medições (pontos). Durante o desenvolvimento da pesquisa, foram encontradas dificuldades tais como: tamanho do problema e número excessivo de variáveis. Desta forma, a complexidade de solução para esse tipo de problema influenciou na escolha de um estudo de caso menor. Optou-se, então, por apresentar o modelo a partir de sete cenários que caracterizam de forma sintética o PRV-IF. O modelo foi executado considerando um período de planejamento de apenas oito dias, contemplando 13 medições.

## 3. Formulação do Modelo Matemático

Neste trabalho, utilizou-se um modelo adaptado do modelo proposto por Cordeau et al. (2001) e citado por Nguyen et al. (2014), para o Problema de Roteamento Periódico de Veículos com Janela de Tempo (PRPVJT). O PRPVJT é uma variação do PRVJT. Nesse modelo, foram realizadas adaptações quanto ao uso de uma única garagem (todos os veículos começam e terminam sua rota na mesma garagem), conforme o trabalho de Mauri (2008).

O modelo proposto para representar o problema de roteamento do planejamento de inventários florestais é descrito a seguir. Os nós *Go* e *Gd* representam, respectivamente, a base de saída e de chegada das equipes na fábrica (garagem). A cada um destes dois nós estão associados os tempos de medição nulos e as janelas de tempo que indicam os dias permitidos de saída e de chegada das equipes à fábrica.

**Minimizar**

$$\sum_{t\in T}\sum_{k\in K}\sum_{i\in\{L_t\cup Go\}}\sum_{j\in\{L_t\cup Gd\}} d_{i,j}\,x^t_{i,j,k} \qquad \forall i\neq j; i=Go \mid j\neq Gd; j=Gd \mid i\neq Go \qquad (1)$$

**Sujeito a:**

$$\sum_{t\in T}\sum_{k\in K}\sum_{j\in\{L_t\cup Gd\}} x^t_{i,j,k}=1 \qquad \forall i\neq j; i=Go \mid j\neq Gd; j=Gd \mid i\neq Go; i\in L_t \qquad (2)$$

$$\sum_{j\in L_t} x^t_{Go,j,k}=1 \qquad \forall k\in K; t\in T \qquad (3)$$

$$\sum_{j\in L_t} x^t_{j,Gd,k}=1 \qquad \forall k\in K; t\in T \qquad (4)$$

$$\sum_{j\in\{L_t\cup Gd\}} x^t_{i,j,k}-\sum_{i\in L_t} x^t_{j,i,k}=0 \qquad \forall i\neq j; i=Go \mid j\neq Gd; j=Gd \mid i\neq Go; k\in K; t\in T \qquad (5)$$

$$H_j \geq (H^t_{Go,k}+d_{Go,j})+M(1-x^t_{Go,j,k}) \qquad \forall k\in K; t\in T; j\in L_t \qquad (6)$$

$$H_j \geq (H_i+s_i+d_{i,j})+M(1-x^t_{i,j,k}) \qquad \forall i\neq j; i=Go \mid j\neq Gd; j=Gd \mid i\neq Go; i\in L_t; j\in\{L_t\cup Gd\} \qquad (7)$$


https://proceedings.science/p/85471?lang=pt-br

$$H^t_{Gd,k} \geq (H_i + s_i + d_{i,Gd})x^t_{i,Gd,k} \qquad \forall k \in K; t \in T; i \in L_t \tag{8}$$

$$H^t_{Go,k} = a \qquad \forall k \in K; t \in T \tag{9}$$

$$H^t_{Gd,k} \leq b \qquad \forall k \in K; t \in T \tag{10}$$

$$x^t_{i,j,k} \in \{0,1\} \qquad \begin{array}{l} \forall i \neq j; i = Go \mid j \neq Gd; j = Gd \mid i \neq Go; \\ k \in K; t \in T; i \in \{L_t \cup Go\}; j \in \{L_t \cup Gd\} \end{array} \tag{11}$$

em que:
$i$ = talhão de origem;
$j$ = talhão de destino;
$Go$ = garagem de origem (representa a fábrica);
$Gd$ = garagem de destino (representa a fábrica);
$d_{i,j}$ = tempo de deslocamento entre os talhões $i$ e $j$;
$d_{Go,j}$ = tempo de deslocamento entre a garagem e o primeiro talhão a ser visitado;
$d_{i,Gd}$ = tempo de deslocamento entre o último talhão visitado e a garagem;
$x^t_{i,j,k}$ = 1 caso o arco ($i$,$j$) for escolhido para integrar a solução para a equipe $k$ no período $t$, e 0 caso contrário;
$T$ = horizonte de planejamento;
$K$ = conjunto de equipes disponíveis;
$L_t$ = lista de talhões (locais) a serem visitados em um período $t$;
$s_i$ = tempo de serviço no nó $i$;
$H_j$ = instante de início do atendimento no talhão $j$;
$H_i$ = instante de início do atendimento no talhão $i$;
$H^t_{Go,k}$ = horário de saída do veículo $k$ no período $t$;
$H^t_{Gd,k}$ = horário de chegada do veículo $k$ no período $t$;
$a$ = início do período de atendimento; e
$b$ = término do período de atendimento.

A função objetivo (1) visa minimizar o deslocamento das equipes de inventário entre talhões. A restrição (2) garante que cada talhão seja visitado uma única vez, por uma única equipe em um determinado período de tempo. A restrição (3) garante que exatamente $k$ equipes sairão da garagem, enquanto a restrição (4) assegura que $k$ equipes cheguem na garagem. Também chamada de "contenção de fluxo", a restrição (5) garante que toda equipe que chega em um talhão, saia dele. A restrição (6) calcula o horário de chegada para atendimento no primeiro talhão, após a equipe deixar a garagem. A restrição (7) garante que o horário de início do atendimento do talhão $j$ seja superior ao tempo de serviço no talhão $i$ somado ao deslocamento entre $i$ e $j$. A restrição (8) calcula o horário de retorno a garagem. As restrições (9) e (10) asseguram que os horários de atendimento ocorram dentro da janela de tempo diária, isto é, entre 0 e 480 minutos. Por fim, a restrição (11) assegura que as variáveis de decisão sejam binárias.

## 4. *Iterated Local Search* (ILS) para resolução do PRV-IF

A meta-heurística ILS, de acordo com Lourenço et al. (2003), é um método de busca iterativa que faz uso de perturbações em ótimos locais com o objetivo de diversificar e visitar outras regiões do espaço de soluções. O pseudocódigo da Figura 3 descreve os quatro principais componentes que definem o método ILS: geração da solução inicial, busca local, perturbação e critério de aceitação.

O primeiro passo do método é construir uma solução inicial $S_0$. Uma vez construída, $S_0$ é submetida a uma busca local onde se obtém $S$* (passos 3 e 4). Posteriormente, enquanto o critério de parada não for atendido, o método realiza iterativamente os seguintes passos: i) efetua uma perturbação na solução corrente $S$* e obtém $S$' (passo 6); ii) aplica uma busca local em $S$' com a finalidade de melhorar a solução perturbada, obtendo assim $S$*' (passo 7); e iii) neste último


https://proceedings.science/p/85471?lang=pt-br

passo, a solução obtida *S**' é submetida ao critério de aceitação, caso aceita, *S**' é retornada 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 3 – Meta-heurística *Iterated Local Search* clássica.

A meta-heurística ILS proposta neste trabalho para resolução do problema de roteamento no planejamento de inventários florestais teve como base o algoritmo proposto por Lourenço et al. (2003). O pseudocódigo da meta-heurística é apresentado na Figura 4. A principal diferença entre o algoritmo proposto e o algoritmo tradicional está na seleção aleatória das estratégias de busca local e de perturbação (passos 4, 14 e 17), realizados a cada iteração do algoritmo. Os movimentos de busca local utilizados foram: 1-Optimal intra-rota e melhor melhora. Para a perturbação, foram utilizados os seguintes movimentos: remoção intra-rota, remoção entre rotas, troca entre rotas e correção. Este último movimento (correção) foi desenvolvido para atender uma necessidade específica do PRV-IF.

```
1.    ILS(TempoMax, d e S)
2.    INÍCIO
3.        S0 ← Construção;
4.        SELECIONAR (aleatoriamente um movimento de Busca Local);
5.        S0 ← BuscaLocal(S0);
6.        S* ← S0;
7.        IncTemp ← 1;
8.        Temp ← TempInicial;
9.        ENQUANTO Tempo ≤ TempoMax FAÇA
10.           SE Temp < TempCongelamento FAÇA
11.               Temp ← TempInicial * IncTemp;
12.           FIM-SE;
13.           PARA (p = 1,2,...,d) FAÇA
14.               SELECIONAR (aleatoriamente um movimento de Perturbação);
15.               S' ← Perturbação(S0);
16.           FIM-PARA;
17.           SELECIONAR (aleatoriamente um movimento de Busca Local);
18.           S' ← BuscaLocal(S');
19.           SE S' < S0 FAÇA
20.               S0 ← S';
21.               SE f(S0) < f(S*) FAÇA
22.                   S* ← S0;
23.               FIM-SE;
24.           SENÃO
25.               SE CriterioAceitacao ENTÃO
26.                   S0 ← S';
27.               FIM-SE;
28.           FIM-SE;
29.           Temp ← Temp * txResfriamento;
30.           IncTemp ← IncTemp + 2;
31.       FIM-ENQUANTO;
32.       S ← S*;
33.   FIM.
```

FIGURA 4 – ILS para o PRV-IF.

### 3.1. Representação de uma Solução

A solução gerada pela meta-heurística considera os seguintes elementos: horizonte de planejamento, número de equipes número de talhões visitados em um determinado dia. Desta forma, a solução é representada por uma matriz de *n* linhas por *m* colunas, sendo *n* = número de


https://proceedings.science/p/85471?lang=pt-br

dias (*H*) x número de equipes (*nk*) e *m*, o número máximo de talhões que podem ser visitados por uma equipe em um dia.

A Figura 5 ilustra um exemplo de representação para um problema com horizonte de 6 dias, 2 equipes e 24 talhões. Observa-se que, cada linha da matriz representa uma rota que uma equipe (*k*) realiza em um dia (*t*). Portanto, a sequência de talhões que deverão ser medidos no dia 5 pela equipe 1 é: 4 – 11 – 20. Os pontos de origem e de destino estão implícitos em cada uma das rotas. Desta forma, no dia 5 a equipe 1 realiza a seguinte rota: 0 – 4 – 11 – 20 – 0, em que 0, representa o ponto de origem e de destino final.

| *t* | *k* | *rotas* | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 8 | | | | |
| 1 | 2 | 7 | | | | |
| 2 | 1 | 15 | 19 | | | |
| 2 | 2 | 24 | | | | |
| 3 | 1 | 16 | | | | |
| 3 | 2 | 23 | 21 | 14 | | |
| 4 | 1 | 12 | | | | |
| 4 | 2 | 5 | 22 | | | |
| 5 | 1 | 4 | 11 | 20 | | |
| 5 | 2 | 17 | 2 | 18 | | |
| 6 | 1 | 13 | 10 | 6 | 3 | 1 |
| 6 | 2 | 9 | | | | |

FIGURA 5 - Representação de uma solução.

Cada talhão está associado a um tempo de serviço ($ts_i$) que corresponde ao produto entre o tempo médio de medição de uma parcela e o número de parcelas. O tempo de deslocamento ($td_{ij}$) entre os pontos (talhões) é dado pela matriz de distância. Portanto, na rota referente ao dia 5 da equipe 1 o tempo total é dado pela soma dos tempos de serviços realizados em 4, 11 e 20, acrescido dos tempos de deslocamento entre 0 – 4, 4 – 11, 11 – 20 e 20 – 0.

### 3.2 Construção da solução inicial

Devido às restrições impostas ao PRV-IF, a solução inicial é construída de maneira aleatória e gulosa, sempre com objetivo de gerar uma solução viável. Porém, em virtude das dificuldades em construir uma solução viável, esta pode ser gerada de maneira inviável, visto que, para cada restrição violada, penaliza-se a Função Objetivo (FO). Este procedimento é apresentado na Figura 6.

```
1.   CONSTRUTIVA(s)
2.   INÍCIO
3.        CRIAR (uma lista T com todos os talhões a serem visitados);
4.        PARA (cada rota l, l = 1,2,...,n, de s) FAÇA
5.             REMOVER (um ponto p de T, que seja compatível com a rota l);
6.             INSERIR (o ponto p na rota l);
7.        FIM-PARA;
8.        PARA (cada ponto p restante de T) FAÇA
9.             REMOVER (um ponto p de T);
10.            CRIAR (uma lista L com todas as rotas possíveis para o ponto p);
11.            PontoIn ← Falso;
12.            PARA (cada rota l de L) FAÇA
13.               SE (pode inserir o ponto p na rota l)
14.                  INSERIR (o ponto p na rota l);
15.                  PontoIn ← Verdadeiro;
16.               FIM-SE;
17.            FIM-PARA;
18.            SE (PontoIn = Falso)
19.                 SELECIONAR (aleatoriamente uma rota l em n);
20.                 INSERIR (o ponto p na rota l);
21.            FIM-SE;
22.       FIM-PARA;
23.  FIM.
```

FIGURA 6 – Pseudocódigo da construção da solução inicial.


https://proceedings.science/p/85471?lang=pt-br

No pseudocódigo apresentado na Figura 6, na primeira iteração (passos 4 a 7), é realizada a inserção de um ponto em cada rota disponível. Essa inserção é importante, visto que há uma restrição no modelo que determina que todos os dias, todas as equipes devem sair da garagem. Na segunda iteração (passos 8 a 22), os pontos restantes, que não foram distribuídos na primeira iteração, são inseridos em rotas escolhidas de maneira restritiva. Essa restrição refere-se ao fato de serem escolhidas apenas rotas em que o ponto pode ser inserido quanto à janela de tempo, e que após inserido, não viole a restrição de janela de tempo diária (480 minutos). Essa verificação é realizada no passo 13 do algoritmo. Caso não seja encontrada uma rota possível, o ponto é inserido em uma rota selecionada aleatoriamente (passos 18 a 21). Neste último caso, a Função Objetivo da solução será penalizada.

### 3.3 Estratégia de perturbação - correção

Diferente dos movimentos citados anteriormente, que são movimentos tradicionais do PRV, o movimento de correção foi desenvolvido para o tipo de problema abordado nesta pesquisa. Neste movimento o objetivo é tornar uma solução viável no que se refere à janela de tempo diária. Para que isso ocorra, pontos são removidos de rotas que violam a restrição de janela de tempo diária, e inseridos em rotas com tempo ocioso. A reinserção dos pontos é priorizada em rotas que não têm nenhum ponto, como forma de corrigir a restrição de que toda equipe deve sair da fábrica diariamente.

O pseudocódigo do movimento de correção é apresentado na Figura 7. Observa-se que, inicialmente são removidos pontos das rotas cujo tempo acumulado seja superior à janela de tempo diária (passos 4 ao 8). A janela de tempo diária é o tempo limite para desenvolvimento das atividades em um dia, que neste trabalho foi de 480 minutos. Os pontos são removidos por ordem de menor impacto, ou seja, pontos com menor tempo de medição.

```
1.  CORREÇÃO(s)
2.  INÍCIO
3.      CRIAR (uma lista T com todos os talhões removidos);
4.      PARA (cada rota l, l = 1,2,...,n, de s) FAÇA
5.          ENQUANTO (tempo acumulado da rota l > janela diária) FAÇA
                REMOVER (um ponto p de menor impacto na rota l);
6.              INSERIR (o ponto p na lista T);
7.          FIM ENQUANTO;
8.      FIM-PARA;
9.      PARA (cada ponto p disponível em T > 0) FAÇA
10.         PontoIn ← Falso;
11.         PARA (cada rota l, l = 1,2,...,n, de s possível para p) FAÇA
12.             SE (tempo acumulado da rota l == 0)
13.                 REMOVER (o ponto p de T);
14.                 INSERIR (o ponto p na rota l);
15.                 PontoIn ← Verdadeiro;
16.             FIM-SE;
17.         FIM-PARA;
18.         SE (PontoIn = Falso)
19.             PARA (cada rota l, l = 1,2,...,n, de s possível para p) FAÇA
20.                 SE (tempo acumulado da rota l < tempo mínimo de ocupação)
21.                     REMOVER (o ponto p de T);
22.                     INSERIR (o ponto p na rota l);
23.                     PontoIn ← Verdadeiro;
24.                 FIM-SE;
25.             FIM-PARA;
26.         FIM-SE;
27.         SE (PontoIn = Falso)
28.             SELECIONAR (aleatoriamente uma rota l em n, possível para p);
29.             INSERIR (o ponto p na rota l);
30.         FIM-SE;
31.     FIM-PARA;
32. FIM.
```

FIGURA 7 – Pseudocódigo do movimento de correção.

Após a remoção, deve-se redistribuir os pontos removidos (disponíveis na lista *T*) para outras rotas. Essa redistribuição é realizada para cada ponto de *T* e ocorre em três etapas. A


https://proceedings.science/p/85471?lang=pt-br

primeira, compreende em redistribuir o ponto para uma das rotas vazias (passos 11 ao 17), como forma de corrigir estas rotas quanto a restrição que determina que todo veículo deve sair da garagem diariamente. Outra restrição também observada é a restrição de janela de tempo (passo 11), visto que existe um período em que ponto deve ser visitado.

Caso não seja encontrada nenhuma rota vazia em que possa ser inserido o ponto, realiza-se uma busca por rotas com uma percentagem de tempo ocioso (passos 19 ao 25). Por fim, caso o ponto ainda não tenha sido inserido em uma das rotas, escolhe-se aleatoriamente uma rota e faz a inserção do ponto (passos 27 ao 30). A Figura 8 ilustra o movimento de correção, em que, a solução inicial é inviável em função do tempo total de serviço (TTS) ser superior à janela de tempo diária (JTD) estabelecida em 480 minutos.

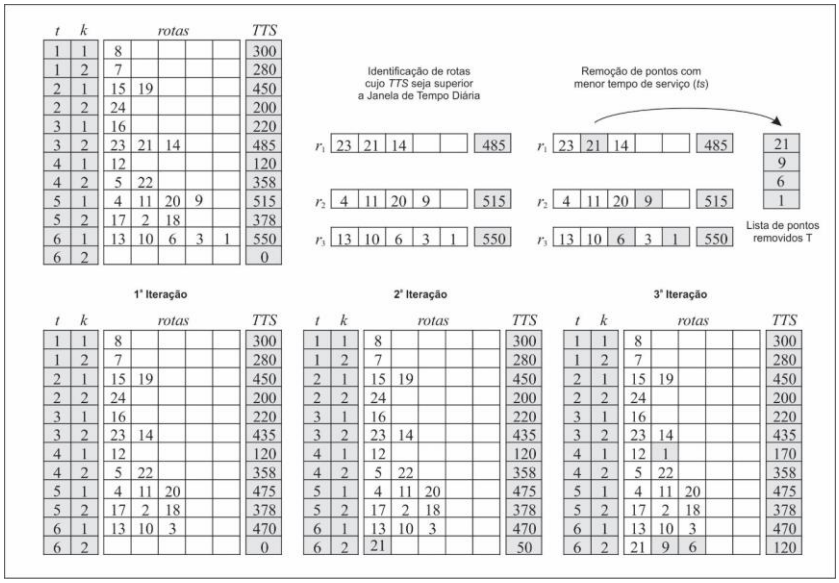



FIGURA 8 – Movimento de correção.

O método seleciona as rotas que inviabilizam a solução ($r_1$, $r_2$ e $r_3$). Para cada uma destas rotas, o movimento remove pontos de menor impacto, ou seja, pontos com menor tempo de serviço, até que o TTS da rota seja menor ou igual à JTD. No exemplo, os pontos 21, 9, 6 e 1 foram removidos das rotas $r_1$, $r_2$ e $r_3$ respectivamente. Observa-se que na rota $r_3$, foi necessária a remoção de dois pontos para que o valor do TTS seja ajustado.

De posse da lista $T$ de pontos removidos, o método reinsere estes pontos em outras rotas obedecendo ao seguinte critério: rotas com menor TTS tem prioridade. Assim, as rotas vazias (vide 1ª iteração e 2ª iteração) têm prioridade por terem seu TTS igual à zero. Isso faz com que a solução seja corrigida quanto à outra restrição do problema que determina que toda equipe deve sair da garagem todos os dias. No exemplo, o ponto 21 foi reinserido na rota que representa o dia 6 da equipe 2.

Inserido os pontos em rotas vazias, os demais são distribuídos nas demais rotas por ordem crescente do TTS. No exemplo, a 3ª iteração ilustra que os pontos 9, 6 e 1 foram inseridos nas rotas de menor TTS. Isso faz com que a restrição de Janela de Tempo Diária seja atendida, tornando a solução viável.

## 4. Experimentos Computacionais

Os experimentos computacionais foram realizados em um conjunto de 9 instâncias elaboradas para esta finalidade. O maior problema contém 2016 pontos (talhões) que corresponde ao número máximo de talhões. Para essa instância foi considerado um horizonte de 252 dias, que representa o número médio de dias úteis em um ano. A partir dessa instância foram geradas as demais conforme representado na Tabela 1. O objetivo foi obter diferentes tamanhos de instâncias para comparação dos métodos. Cada instância está associada a um número de equipes

e a um tempo de processamento. Este último foi estabelecido com base no tempo gasto pelo método exato para encontrar uma solução ótima para a menor instância em segundos.

Tabela 1- Instâncias para o PRV-IF.

| Instância | Nº de Pontos | Horizonte | Nº Equipes | Tempo |
|---|---|---|---|---|
| PRV-IF_0024_006 | 24 | 6 dias | 2 | 120'' |
| PRV-IF_0050_012 | 50 | 12 dias | 2 | 180'' |
| PRV-IF_0100_024 | 100 | 24 dias | 2 | 240'' |
| PRV-IF_0100_036 | 100 | 36 dias | 2 | 240'' |
| PRV-IF_0200_048 | 200 | 48 dias | 2 | 480'' |
| PRV-IF_0400_060 | 400 | 60 dias | 2 | 960'' |
| PRV-IF_0500_060 | 500 | 60 dias | 3 | 1200'' |
| PRV-IF_1000_120 | 1000 | 120 dias | 3 | 2400'' |
| PRV-IF_2016_252 | 2016 | 252 dias | 3 | 4800'' |

Para ajustar os parâmetros do método ILS, foram escolhidas três instâncias ao acaso: PRV-IF_0024_006, PRV-IF_0100_024 e PRV-IF_0200_048. Foram realizadas 5 execuções para cada instância com cada combinação. Embora o ILS tenha apenas um parâmetro *d* que representa a taxa de destruição, foram obtidas três faixas de valores para instâncias acima de 500 pontos, entre 500 e 100 pontos e menor ou igual a 100 pontos. A faixa de valores utilizada para o parâmetro *d* foi [0,001, 0,025, 0,050]. Os melhores valores encontrados para os parâmetros são apresentados na Tabela 2.

Tabela 2 - Valores obtidos para o parâmetro do método ILS.

| Meta-heurística | Parâmetro | Significado | Melhor valor |
|---|---|---|---|
| ILS | *d* | Taxa de destruição > 500 pontos | 0,001 |
| | *d* | Taxa de destruição ≥ 100 ≤ 500 pontos | 0,025 |
| | *d* | Taxa de destruição < 100 pontos | 0,050 |

Os testes exatos foram executados no *software* CPLEX em uma máquina com microprocessador Intel Xeon 2.40 GHz com 32 GB de memória RAM utilizando o sistema operacional Windows 10. A meta-heurística ILS foi desenvolvida na linguagem de programação C++. Os testes foram efetuados em uma máquina com microprocessador Intel Core i5 750 2.67 GHz com 8 GB de memória RAM utilizando o sistema operacional Windows 7.

Nesta pesquisa foi proposta a meta-heurística ILS como alternativa de solução do PRV-IF. Para obter os resultados médios dessa meta-heurística, executou-se 30 vezes cada método para cada instância. Foi estabelecido como condição de parada o tempo de processamento. Em todas as execuções foram encontradas soluções viáveis. Estes resultados são comparados aos resultados obtidos pelo *software* CPLEX.

Os experimentos computacionais são resumidos na Tabela 3. Para cada instância, são apresentados os valores obtidos pelos métodos CPLEX e 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).

Com base na análise dos resultados da Tabela 3, pode-se observar que o método exato foi capaz de obter a solução ótima apenas para a menor instância (PRV-IF_0024_006). Para as instâncias PRV-IF_0050_012 e PRV-IF_0100_036, embora o CPLEX tenha encontrado soluções exatas, estas ficaram distantes da solução ótima em no máximo 5,88% e 8,32% respectivamente (vide coluna GAP). Nas instâncias de PRV-IF_0100_024 e PRV-IF_0200_048, o CPLEX não foi capaz de encontrar soluções ótimas, indicando apenas os limitantes inferiores. Nas demais instâncias, o CPLEX não foi capaz de encontrar soluções.

A meta-heurística ILS foi capaz de encontrar soluções viáveis para 8 das 9 instâncias

testadas. Apenas na instância PRV-IF_2016_252, a meta-heurística não foi capaz de encontrar solução viável no tempo estabelecido. Neste caso, a janela de tempo diária (JTD) desta instância foi flexibilizada na tentativa de encontrar soluções viáveis para o problema. Desta forma, utilizou-se a JTD em 490 minutos (instância PRV-IF_2016_252[1]), 500 minutos (instância PRV-IF_2016_252[2]) e 510 minutos (instância PRV-IF_2016_252[3]). Com essa flexibilização, o método foi capaz de encontrar soluções viáveis. Nestes cenários, a empresa pode avaliar o custeio das horas extras em relação ao ganho de tempo no desenvolvimento das atividades.

A meta-heurística ILS obteve bons resultados quando comparados com o CPLEX. Apenas nos problemas PRV-IF_0050_012 e PRV-IF_0100_036 o CPLEX foi superior ao ILS. O bom desempenho deste método também pode ser verificado quando se avalia o coeficiente de variação, visto que em apenas uma situação ele ultrapassou a margem de um por cento. Isso ocorreu com a instância PRV-IF_0050_012. Os melhores valores encontrados são apresentados em negrito.

Tabela 3 - Resultados obtidos pelos métodos CPLEX e ILS para as instâncias do PRV-IF.

| Instância | CPLEX | | | | ILS | | | |
|---|---|---|---|---|---|---|---|---|
| | FO | Limitante | Tempo | GAP | FO | Média | CV(%) | Tempo |
| PRV-IF_0024_006 | 132,61 | 132,61 | 43,78 | 0.00 | **132,61** | 133,12 | 0,36 | 2,60 |
| PRV-IF_0050_012 | **2318,97** | 2190,03 | 1787,20 | 5.88 | 2418,33 | 2522,96 | 1,12 | 141,39 |
| PRV-IF_0100_024 | - | 3722,80 | 357,22 | ∞ | **4606,25** | 4672,29 | 0,81 | 148,18 |
| PRV-IF_0100_036 | **6446,30** | 5951,28 | 47103,37 | 8.32 | 6481,70 | 6596,90 | 0,72 | 16,99 |
| PRV-IF_0200_048 | - | 6307,32 | 6,44 | ∞ | **11356,34** | 11634,67 | 0,91 | 234,75 |
| PRV-IF_0400_060 | - | - | - | - | **23806,79** | 24137,14 | 0,59 | 581,73 |
| PRV-IF_0500_060 | - | - | - | - | **24673,59** | 25192,68 | 0,86 | 283,34 |
| PRV-IF_1000_120 | - | - | - | - | **63245,32** | 64202,35 | 0,61 | 2045,55 |
| PRV-IF_2016_252[1] | - | - | - | - | **126627,15** | 128696,41 | 0,64 | 4766,06 |
| PRV-IF_2016_252[2] | - | - | - | - | **124472,54** | 126013,76 | 0,60 | 3189,07 |
| PRV-IF_2016_252[3] | - | - | - | - | **125023,11** | 127110,10 | 0,80 | 4468,10 |

A coluna FO (melhor valor para a Função Objetivo encontrado) representa o tempo de deslocamento das equipes em minutos. 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 do método ILS. A coluna CV representa o coeficiente de variação da média.

## 5. Conclusões

Nesta pesquisa, buscou-se propor um método capaz de encontrar soluções para o problema de roteamento de equipes no planejamento de inventários florestais. Foi utilizado um modelo adaptado do Problema de Roteamento Periódico de Veículos com Janela de Tempo. Os métodos de solução utilizados foram o método exato por meio do software CPLEX e a meta-heurística ILS.

A modelagem do problema contribuiu de forma eficaz na obtenção dos resultados, uma vez que toda a programação de rotas estabelecidas para cada equipe e em cada dia de planejamento foi obtida. Além da definição das rotas, também foram estabelecidos os horários de saída e chegada de cada equipe nos talhões e na fábrica.

Os experimentos computacionais mostraram um bom desempenho da meta-heurística em relação aos resultados encontrados pelo solver CPLEX. Foram obtidos avanços na metodologia em relação ao trabalho proposto por Meneguzzi (2011), em especial na obtenção de rotas diárias e uso de um método capaz de obter soluções viáveis para problemas em escalas muito superiores.

## Agradecimentos

Os autores agradecem a empresa Fibria pela conceção dos dados para o desenvolvimento desta pesquisa. Os autores também agradecem ao Conselho Nacional de Desenvolvimento

Científico e Tecnológico – CNPq (processos 454569/2014-9 and 301725/2016-0) e Fundação de Amparo à Pesquisa e Inovação do Espírito Santo – FAPES (processo 73290475/2016).

**Referências**

**Arenales, M. et al.** *Pesquisa Operacional para os Cursos de Engenharia*. 2. ed. Rio de Janeiro: Elsevier, 2015. 564 p.

**Campos, J. C. C. e Leite, H. G.** *Mensuração florestal: perguntas e respostas*. 5. ed. Viçosa: UFV, 2017. 632 p.

**Cordeau, J. -F., Laporte, G. e Mercier, A.** (2001), A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*. 52, 8, 928–936.

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

**Goldbarg, E., Goldbarg, M. C. e Luna, H. P. L.** *Otimização Combinatória e Meta-heurísticas*: Algoritmos e Aplicações, Rio de Janeiro: Campus, 2015. 416 p.

**Haddad, H. M. D.** Roteamento otimizado no inventário florestal das árvores de Belo Horizonte. 2015, 63 f. *Dissertação de Mestrado em Engenharia Florestal* – Programa de Pós-Graduação em Engenharia Florestal, Universidade Federal de Lavras, Lavras, 2015.

**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.

**Mauri, G. R.** Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória. 2008, 241 f. *Tese de Doutorado em Computação Aplicada* – Programa de Pós-Graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2008.

**Marcatti, G. E.** Caminhamento ótimo para acesso às parcelas de inventário florestal. 2013, 32f. *Dissertação de Mestrado em Ciência Florestal* – Programa de Pós-Graduação em Ciência Florestal, Universidade Federal de Viçosa, Viçosa, 2013.

**Meneguzzi, C. C.** Modelo de roteamento de veículos aplicado ao planejamento do inventário florestal. 2011, 94 f. *Dissertação de Mestrado em Ciências Florestais* – Programa de Pós-Graduação em Ciências Florestais, Universidade Federal do Espírito Santo, Jerônimo Monteiro, 2011.

**Nguyen, P. K., Crainic, T. G. e Toulouse, M.** (2014), A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. *Journal of Heuristics*, 20, 4, 383–416.

**Soares, C. P. B. et al.** *Dendrometria e inventário florestal*. Viçosa: UFV, 2011, 272 p.

**Toth, P. e Vigo, D.** *Metaheuristics for the VRP. In: Toth, P. e Vigo, D. (Ed.)*. The vehicle routing problem. Philadelphia: SIAM, 2002, p. 129-154.

**Vieira, H. P.** Metheuristica para a solução de problemas de roteamento de veículos com janela de tempo. *Dissertação de Mestrado em Matemática Aplicada* – Instituto de Matemática, Estatística e Computação Científica - UNICAMP, Campinas, 2013.