

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE  
DEPARTAMENTO DE COMPUTAÇÃO**

**DENIS PAULA BITTENCOURT TOSE**

**APLICAÇÃO DA META-HEURÍSTICA CLUSTERING  
SEARCH NA PROGRAMAÇÃO DE ROTAS PARA EQUIPES  
DE INVENTÁRIO FLORESTAL**

**ALEGRE**

**2016**

DENIS PAULA BITTENCOURT TOSE

**APLICAÇÃO DA META-HEURÍSTICA CLUSTERING  
SEARCH NA PROGRAMAÇÃO DE ROTAS PARA EQUIPES  
DE INVENTÁRIO FLORESTAL**

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. M.Sc. Antonio Almeida de Barros Junior.

ALEGRE

2016

**DENIS PAULA BITTENCOURT TOSE**

**APLICAÇÃO DA META-HEURÍSTICA CLUSTERING  
SEARCH NA PROGRAMAÇÃO DE ROTAS PARA EQUIPES  
DE INVENTÁRIO FLORESTAL**

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em \_\_\_\_ de novembro de 2016.

**COMISSÃO EXAMIDADORA**

---

**Prof. M.Sc Antonio Almeida de Barros Junior**  
**Universidade Federal do Espírito Santo**  
**Orientador**

---

**Prof. Dr. Geraldo Regis Mauri**  
**Universidade Federal do Espírito Santo**

---

**Prof. M.Sc. Dayan de Castro Bissoli**  
**Universidade Federal do Espírito Santo**

*“Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que as outras, enquanto o imenso oceano da verdade continua misterioso diante de meus olhos”.*

**Isaac Newton**

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pelas oportunidades, por ter me guiado até aqui e me dado forças nos momentos onde estive desacreditado em mim mesmo e perdido em relação aos meus objetivos.

A minha família, especialmente a minha mãe Leni Bittencourt de Aquino por todos os ensinamentos, por fazer de mim uma pessoa integra, por todo carinho e amor dado a mim. Ao meu padrasto Paulo Roberto Lino de Aquino e ao meu pai Valdecir Tose, por todo apoio nas horas difíceis, pelos bons conselhos e por não medirem esforços para me proporcionar mais essa conquista.

Aos meus amigos que sempre estiveram ao meu lado durante esta jornada, em especial ao Wagner Amorim pela paciência em me ajudar e pela motivação para seguir sempre em frente; ao João Vitor Costa e Rodrigo Torres por sempre estarem ao meu lado durante esses anos, pelos conselhos e pela companhia; ao Josimar Moraes como companheiro de república, e ao Jefferson Moraes por sempre preparar um bom café para as noites de estudo.

A Ana Paula, Caíque, Marília, Melissa e a galera do SDC pelas noites de estudo, de baralho, por aguentarem minhas reclamações e sempre me aconselharem, pelos grandes momentos compartilhados de muitas risadas de muito café de muitas séries, muito RU e de muitas zoeiras.

Ao meu orientador Antonio Almeida de Barros Junior pelos ensinamentos, pela paciência com meus erros e pela oportunidade que me foi dada de participar deste projeto.

A Universidade Federal do Espírito Santo e todos seus funcionários, em especial aos professores do departamento de computação por me transmitirem o conhecimento e experiências vividas. Também meu agradecimento à universidade pelo apoio financeiro aos dois projetos de monitoria em que tive oportunidade de trabalhar.

A empresa Click Junior pelos quase três anos de trabalho, pela experiência que tive oportunidade de adquirir, por me permitir testar os conhecimentos obtidos durante os anos de graduação e pelos bons amigos que fiz na empresa.

A todos, muito obrigado.

## RESUMO

O inventário florestal é uma atividade de coleta de dados pertencente à fase de planejamento do empreendimento florestal, sendo que suas informações auxiliam a organização possibilitando uma tomada de decisão mais segura. Este trabalho tem como objetivo otimizar a alocação das equipes responsáveis pela coleta de dados, utilizando um modelo do Problema de Roteamento Periódico de Veículos com Janela de Tempo (PRPVJT). Para resolver o Problema de Alocação de Equipes de Inventário Florestal (PAEIF), propõem-se a utilização da meta-heurística híbrida *Clustering Search* (CS), empregando como gerador de soluções o método *Simulated Annealing* (SA). Nesse trabalho foram realizados testes utilizando instâncias de diferentes tamanhos para aplicação da CS. As instâncias de entrada foram geradas a partir de dados reais, previamente fornecidos por uma empresa atuante no mercado de produtos florestais. A meta-heurística CS mostrou-se eficiente na resolução do PAEIF, e quando comparada ao *software* CPLEX, a CS obteve melhora para a maioria das soluções conhecidas, com um baixo tempo de execução.

Palavras-chave: *Simulated Annealing*; Problema de Roteamento Periódico de Veículos com Janela de Tempo; Problema e Alocação de Equipes de Inventário Florestal;

## **ABSTRACT**

The forest inventory is a data collection activity belonging to the planning stage of the forest enterprise, and its information provides support to the organization for safer decision making. This work aims to optimize the allocation of the teams responsible for data collection, using a Model of the Periodic Routing of Vehicles with Time Window (PRPVJT). In order to solve the Problem and Allocation of Forest Inventory Teams (PAEIF), we propose the use of the hybrid meta-heuristic Clustering Search (CS), using Simulated Annealing (SA) as the generator of solutions. In this work, tests were performed using different size instances for CS application. The input instances were generated from actual data previously provided by a company active in the forest products market. The CS meta-heuristic proved to be efficient in the resolution of the PAEIF, and when compared to the CPLEX software, the CS obtained improvement for most known solutions with a low execution time.

Keywords: Simulated Annealing; Problem of Periodic Routing of Vehicles with Time Window; Problem and Allocation of Forest Inventory Teams;

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 - Representação de uma de solução com 24 talhões e 6 dias.....      | 22 |
| Figura 2 - Matriz Solução antes do movimento Troca Sequência de Visita. .... | 23 |
| Figura 3 - Matriz Solução depois do movimento Troca Sequência de Visita..... | 24 |
| Figura 4 - Matriz Solução antes do movimento Troca Talhão de Equipe. ....    | 25 |
| Figura 5 - Matriz Solução antes do movimento Troca Talhão de Equipe. ....    | 25 |
| Figura 6 - Matriz Solução antes do movimento Troca Talhão de Dia.....        | 26 |
| Figura 7 - Matriz Solução depois do movimento Troca Talhão de Dia. ....      | 27 |
| Figura 8 - Exemplo de matriz de janela de tempo para o talhão 15. ....       | 27 |
| Figura 9 - Fluxograma da CS. Fonte: Chaves e Lorena (2010).....              | 29 |
| Figura 10 - Pseudocódigo da SA.....  | 30 |
| Figura 11 - Exemplo do cálculo da distância de Hamming.....                  | 31 |
| Figura 12 - Pseudocódigo da CS utilizando SA.....                            | 33 |



## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 - Representação da Quantidade de Clientes.....                | 22 |
| Tabela 2 - Valores das penalizações. ....                              | 34 |
| Tabela 3 – Combinação de valores para os parâmetros.....               | 34 |
| Tabela 4 - Valores dos parâmetros. ....                                | 35 |
| Tabela 5 - Resultados de dez execuções do CS para cada movimento. .... | 36 |
| Tabela 6 – Comparação com as soluções obtdas pelo CPLEX. ....          | 37 |

## SUMÁRIO

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>INTRODUÇÃO</b>   | 12 |
| 1.1      | O PROBLEMA E SUA IMPORTÂNCIA  | 12 |
| 1.2      | OBJETIVOS   | 14 |
| 1.2.1    | <b>Objetivo geral</b>   | 14 |
| 1.2.2    | <b>Objetivos específicos</b>  | 14 |
| 1.3      | ESTRUTURA DO TRABALHO   | 14 |
| <b>2</b> | <b>REVISÃO DE LITERATURA</b>  | 15 |
| 2.1      | INVENTÁRIO FLORESTAL  | 15 |
| 2.2      | O PROBLEMA DE ROTEAMENTO DE VEÍCULOS                                    | 15 |
| 2.2.1    | <b>Problema de roteamento periódico de veículos com janela de tempo</b> | 16 |
| 2.2.2    | <b>Modelo do PRPVJT para o PAEIF</b>                                    | 17 |
| 2.3      | META-HEURÍSTICA CS PARA O PRV   | 20 |
| <b>3</b> | <b>METODOLOGIA</b>  | 21 |
| 3.1      | REPRESENTAÇÃO DE UMA SOLUÇÃO  | 21 |
| 3.2      | ESTRUTURAS DE VIZINHANÇA  | 22 |
| 3.2.1    | <b>Movimento Troca Sequência de Visita</b>                              | 23 |
| 3.2.2    | <b>Troca Talhão de Equipe</b>   | 24 |
| 3.2.3    | <b>Troca Talhão de Dia</b>  | 25 |
| 3.3      | CLUSTERING SEARCH   | 27 |
| 3.3.1    | <b>Geradora de Solução</b>  | 29 |
| 3.3.2    | <b>Agrupamento Iterativo</b>  | 30 |

|              |                                 |           |
|--------------|---------------------------------|-----------|
| <b>3.3.3</b> | <b>Busca local</b> .....        | <b>31</b> |
| <b>3.3.4</b> | <b>Função objetivo</b> .....    | <b>32</b> |
| <b>3.3.5</b> | <b>Pseudocódigo da CS</b> ..... | <b>33</b> |
| <b>4</b>     | <b>RESULTADOS</b> .....         | <b>34</b> |
| <b>5</b>     | <b>CONCLUSÕES</b> .....         | <b>39</b> |
| <b>6</b>     | <b>REFERÊNCIAS</b> .....        | <b>40</b> |

## 1 INTRODUÇÃO

O planejamento florestal é composto por atividades estratégicas que tendem a indicar os procedimentos corretos a serem adotados para manutenção das florestas. Nesse contexto, o inventário florestal surge como um importante procedimento para quantificação e avaliação do povoamento florestal.

Atividades ligadas ao planejamento florestal, em sua maioria, exigem um esforço contínuo e elevados custos associados a coleta de dados (CAMPOS; LEITE, 2006). Nesse sentido, acentua-se a necessidade de utilizar modelos já conhecidos na Otimização Combinatória (OC) para minimizar a distância percorrida pelas equipes que realizam a coleta de dados. Em problemas de OC o espaço de possíveis soluções tende a ser finito, facilitando a utilização de seus elementos para a realização de testes em busca de possíveis soluções para o problema tratado.

Em problemas de OC, testar todas as possibilidades de um problema pode ser inviável computacionalmente. Encontrar soluções viáveis para estes problemas também pode se tornar uma tarefa exaustiva e até mesmo impraticável, em alguns casos. Dessa forma é justificável abrir mão da otimalidade das soluções em busca de boas aproximações, encontradas de forma eficiente (CARVALHO *et al.*, 2001). Optar por soluções aproximadas é o paradigma dos algoritmos de aproximação que são utilizados neste trabalho para resolver o Problema de Alocação de Equipes de Inventário Florestal (PAEIF).

### 1.1 O PROBLEMA E SUA IMPORTÂNCIA

Dentre as diversas atividades que envolvem um empreendimento florestal, as que auxiliam na fase de planejamento necessitam de uma contínua busca por melhorias. A atividade de Inventário Florestal é considerada uma das mais importantes por se tratar de uma atividade primária no processo de tomada de decisão. Nessa atividade, os dados são coletados nas florestas periodicamente, com o objetivo de avaliar o desenvolvimento da população florestal e auxiliar no planejamento e manejo da produção.

De acordo com Campos e Leite (2006), a coleta de dados para compor o inventário florestal é responsável por grande parte dos custos relacionados à atividade. Diretamente associadas aos custos de produção, as tarefas que proveem dados para o inventário florestal normalmente envolvem um grande número de variáveis, transformando os mínimos erros cometidos no planejamento em grandes prejuízos.

Para um melhor reconhecimento da população florestal, as áreas a serem visitadas para coleta de dados são divididas em fazendas, e estas são subdivididas em talhões. Cada talhão pode conter uma ou mais parcelas (unidades amostrais onde os dados são coletados) que devem ser visitadas em períodos de tempo previamente conhecidos.

A cada ano é realizado um planejamento com o objetivo de definir quais equipes irão inventariar quais talhões em que período. O período de medição de um talhão é determinado pela idade de plantio do povoamento. Esse planejamento, também denominado de programação anual de atividades de inventário florestal, é realizado no escritório e repassado às equipes de inventário para execução.

No desenvolvimento da programação anual, o fator distância entre os talhões muitas vezes não é levado em consideração. Isso determina que equipes de inventário tenham que retornar a uma mesma região diversas vezes em período curto de horizonte de planejamento. Com objetivo de otimizar o deslocamento das equipes de inventário florestal, Meneguzzi (2011) propôs um modelo de roteamento de equipes baseado no Problema de Roteamento de Veículos (PRV). Nesse trabalho, a autora não conseguiu grandes avanços no que se refere ao número de talhões, utilizando o *software* CEPLEX para solução do problema.

Fundamentado no trabalho de Meneguzzi (2011), Barros Junior (2016) propôs um modelo de roteamento de veículos para equipes de inventário florestal. Esse modelo baseia-se no Problema de Roteamento Periódico de Veículos com Janelas de Tempo, no qual o conceito cliente é substituído por talhão e o conceito de veículo é substituído por equipe.

Com base no que fora exposto, este trabalho apresenta a aplicação da meta-heurística híbrida *Clustering Search* (CS) como alternativa para solucionar o Problema de Alocação de Equipes de Inventário Florestal (PAEIF).

## 1.2 OBJETIVOS

A seguir são descritos os objetivos gerais e específicos relacionados a este trabalho.

### 1.2.1 Objetivo geral

O objetivo deste trabalho foi desenvolver e testar a meta-heurística *Clustering Search* (CS) como alternativa para programação de rotas de equipes de inventário florestal. O desempenho dessa meta-heurística foi comparado com resultados obtidos por um método exato na resolução do problema.

### 1.2.2 Objetivos específicos

- a) Implementar a estrutura para representar o Problema de Alocação de Equipes de Inventário Florestal (PAEIF).
- b) Implementar a meta-heurística *Clustering Search* utilizando o *Simulated Annealing* para gerar soluções;
- c) Avaliar o desempenho da meta-heurística, comparando os resultados com os obtidos pelo *software* CEPLEX;

## 1.3 ESTRUTURA DO TRABALHO

O restante deste trabalho está dividido como segue. No Capítulo 2 é apresentada uma revisão bibliográfica dos principais e mais recentes trabalhos da literatura. O Capítulo 3 detalha a metodologia utilizada. Os resultados obtidos são descritos no Capítulo 4, as conclusões são apresentadas no Capítulo 5, e no Capítulo 6 são listadas as referências.

## 2 REVISÃO DE LITERATURA

### 2.1 INVENTÁRIO FLORESTAL

O inventário florestal é um procedimento que provê à organização informações sobre quantidade e qualidade dos recursos florestais. O mesmo auxilia na definição das características referentes às áreas onde as árvores são cultivadas (HUSH et al., 2003).

De acordo com Meneguzzi (2011), a atividade de inventário florestal pode ser dividida em etapas, dentre estas, a etapa de amostragem abrange a fase de coleta dos dados nas florestas, é o momento onde as equipes previamente alocadas visitam os talhões de coleta. Outra etapa que compõe atividade de inventário florestal é o mapeamento da área a ser amostrada, é um fator de sucesso para o inventário florestal, sua importância está associada a necessidade de utilizar as informações obtidas na etapa de amostragem em todo o povoamento florestal.

Soares et al. (2007) definiram um *checklist* com os itens a serem considerados na elaboração do planejamento de um inventário florestal. Dentre esses, o item “procedimentos para o trabalho de campo” apresenta uma série de pontos a serem observados pelo engenheiro florestal. O suporte logístico e de transporte e a alocação de equipes estão entre os fatores que podem ser relevantes para um inventário florestal por amostragem. Nesse contexto, o Problema de Roteamento de Veículos (PRV) é considerada uma abordagem válida para a alocação de equipes

### 2.2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS

O Problema de Roteamento de veículo (PRV) é proveniente do clássico problema do caixeiro viajante, no qual um caixeiro precisa visitar um conjunto de cidades de forma a minimizar a distância percorrida entre as visitas. O caixeiro viajante sai de uma cidade base ou de um depósito e deve visitar todas as cidades do conjunto estabelecido inicialmente, passando por cada cidade uma única vez. Os problemas do tipo caixeiro viajante pertencem a classe de roteamento em nós. Dessa forma, podem ser definidos com grafos, orientados ou não orientados. (ARENALES et. al., 2007; LOESCH; HEIN, 2009).

De acordo com Arenales et al. (2007), o PRV está relacionado a definição de rotas para entrega e/ou coleta de recursos. Projetar as rotas envolve uma função objetivo

que trata a minimização de custos. Uma definição em forma de grafo para representar os nós clientes e os percursos entre eles pode ser apresentada, como segue. Seja um grafo  $G = (N, E)$  em que  $N = C \cup \{0, n + 1\}$ ,  $C = \{1, \dots, n\}$  é o conjunto de nós clientes, e os nós 0 e  $n + 1$  representam o depósito. Para representar os arcos (arestas) de conexão entre os nós (vértices) é possível definir o seguinte conjunto  $E = \{(i, j) : i, j \in N, i \neq j, i \neq n + 1, j \neq 0\}$ . Dessa forma,  $c_{ij}$  representa o custo de descolamento de um vértice  $i$  para um vértice  $j$  do grafo. Nenhuma rota poderá iniciar a partir do nó  $n + 1$  ou se encerrar no vértice 0 do grafo.

Segundo Golden et al. (1995), o PRV pertence à classe de problemas NP-difícil. Portanto, as meta-heurísticas são as técnicas mais utilizadas para resolução do problema. Para o PRV, a literatura indica diversos métodos à serem utilizados, dentre os quais destacam-se os Algoritmos Genéticos (NGUYEN *et al.*, 2011), GRASP (FEO; RESENDE, 1995), Busca Tabu (CORDEAU *et al.*, 2011), entre outros.

### 2.2.1 Problema de roteamento periódico de veículos com janela de tempo

Segundo Pillac et al. (2013), o PRV possui algumas variações comumente encontradas em problemas específicos na literatura. Tais variações se dão pelo fato de diversas limitações e regras adicionais serem impostas ao modelo já conhecido do PRV em aplicações reais. A seguir, são descritas algumas das principais variações existentes para o PRV:

- Problema de Roteamento de Veículos Capacitado – PRVC;
- Problema de Roteamento de Veículos com Janela de Tempo – PRVJT;
- Problema de Roteamento de Veículos com Frota Heterogênea – PRFH;
- Problema de Roteamento de Veículos com Coleta e Entrega simultânea – PRVCES;
- Problema de Roteamento de Veículos Dinâmico – PRVD;
- Problema de Roteamento Periódico de Veículos – PRPV;
- PRPVJT – Problema de Roteamento Periódico de Veículos com Janela de Tempo; entre outros.

O modelo de roteamento utilizado nesta pesquisa tem características semelhantes ao Problema de Roteamento Periódico de Veículos com Janelas de Tempo (PRPVJT).



No PRPVJT cada talhão a ser visitado possui o seu ciclo de tempo para medição e consequentemente intervalos distintos para medição durante o ciclo de vida da floresta. O problema também visa o planejamento diário das visitas, bem como a alocação das equipes diariamente.

Segundo Nguyen et al. (2011), o PRPVJT é definido em um grafo não-dirigido completo  $G = (V, E)$ , em que  $V = \{0, 1, \dots, n\}$  é o conjunto de vértices e  $E = \{(i, j) : i, j \in V, i \neq j\}$  é o conjunto de arestas. Uma distância (ou tempo de percurso)  $C_j$  está associado com cada aresta  $(i, j) \in E$ . O vértice que representa o depósito é indicado por 0.  $V_C$  é o conjunto de vértices (clientes). Cada vértice  $i \in V_C$  tem uma demanda  $Q_i > 0$  em cada dia do horizonte de planejamento de  $T$  dias, um tempo de serviço  $S_i > 0$  e uma janela de tempo  $[e_i, l_i]$ . O instante de tempo inicial da janela em que o vértice  $V_i$  pode ser atendido é representado por  $e_i$  e o instante de tempo final da janela em que  $V_i$  poderá ser atendido é representado por  $l_i$ . Cada vértice  $V_i$  requer um número fixo de visitas  $f_i$ , a serem realizadas de acordo com um dos padrões de visitas diárias admissíveis em uma lista  $R_i$ . A janela de tempo específica para o intervalo de saída e chegada dos veículos ao depósito é dada por  $[e_0, l_0]$ .

### 2.2.2 Modelo do PRPVJT para o PAEIF

Barros Junior (2016) definiu uma formulação matemática mais específica para o PAEIF. A modelagem sugerida descreve as restrições impostas para uma aplicabilidade real do modelo na representação do problema. Dentre as adaptações feitas aos modelos tradicionais de PRPVJT, o autor optou por remover a restrição relacionada a capacidade dos veículos considerando que a atividade de inventário florestal não prioriza o transporte de cargas.

Os nós  $G_o$  e  $G_d$  da formulação matemática representam, respectivamente, a base de saída e de chegada das equipes na fábrica (depósito). A cada um desses dois nós estão associados tempos de medição nulos e janelas de tempo que indicam os dias permitidos de saída e de chegada das equipes à fábrica.

A formulação matemática para o problema, de acordo com Barros Junior (2016) é dada por:

**Minimizar**

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

**Sujeito a:**

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

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

$$\sum_{i \in L_t} x_{j,Gd,k}^t = 1 \quad \forall k \in K; t \in T \quad (4)$$

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

$$H_j \geq (H_{Go,k}^t + d_{Go,j}) + M(1 - x_{Go,j,k}^t) \quad \forall k \in K; t \in T \quad (6)$$

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

$$H_{Gd,k}^t \geq (H_i + s_i + d_{i,Gd}) x_{i,Gd,k}^t \quad \forall k \in K; t \in T \quad (8)$$

$$H_{Go,k}^t = a \quad \forall k \in K; t \in T \quad (9)$$

$$H_{Gd,k}^t \leq b \quad \forall k \in K; t \in T \quad (10)$$

$$x_{i,j,k}^t \in \{0,1\} \quad \forall i \neq j; i = Go \mid j \neq Gd; j = Gd \mid i \neq Go; k \in K; t \in T; i \in \{L_t \cup Go\}; j \in \{L_t \cup Gd\} \quad (11)$$

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;

$x_{i,j,k}^t = 1$  caso o arco  $(i, j)$  seja 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  $i$ ;

$H_{Go,k}^t$  = horário de saída da equipe  $k$  no período  $t$ ;

$H_{Gd,k}^t$  = horário de chegada da equipe  $k$  no período  $t$ ;

$a$  = início do período de atendimento do talhão;

$b$  = término do período de atendimento do talhão;

$M$  = constante suficientemente grande.

A função objetivo (1) visa minimizar o deslocamento das equipes de inventário entre talhões. As restrições (2) garantem que cada talhão seja visitado uma única vez, por uma única equipe em um determinado período de tempo. As restrições (3) garantem que exatamente  $k$  equipes saíam da garagem, enquanto as restrições (4) asseguram que  $k$  equipes cheguem na garagem. Também chamadas de “contenção de fluxo”, as restrições (5) garantem que toda equipe que chega em um talhão, saia dele. As restrições (6) calculam o horário de chegada para atendimento no primeiro talhão, após a equipe deixar a garagem. As restrições (7) garantem que horário de início do atendimento do talhão  $j$  seja superior ao tempo de serviço no talhão  $i$  somado ao deslocamento entre  $i$  e  $j$ . As restrições (8) calculam o horário de retorno a garagem. As restrições (9) e (10) asseguram que os horários de atendimento ocorram dentro da

janela de tempo diária. Ou seja, entre 0 e 480 minutos. E por fim, as restrições (11) asseguram que as variáveis de decisão sejam binárias.

### 2.3 META-HEURÍSTICA CS PARA O PRV

A meta-heurística *Clustering Search* (CS) apresentou resultados satisfatórios quando utilizada por Araújo et al. (2013) no problema de roteamento periódico de veículos para coleta de lixo. O método melhorou os resultados atualmente conhecidos e até mesmo reduzir a quantidade de rotas realizadas para coleta de lixo, utilizando instâncias reais da cidade portuguesa Ponte de Lima. Através da CS, foi possível reduzir em 30% a distância total percorrida pelos caminhões de coleta, quando considerada a solução adotada atualmente no município.

De acordo com Araújo et al. (2013), o método gerador de soluções *Simulated Annealing* (SA) sem o processo de agrupamento obteve soluções piores em termos qualitativos em relação à CS. A melhor solução encontrada pela CS é aproximadamente 1% melhor, em termos qualitativos, que a melhor solução encontrada pelo SA. Os autores consideraram a CS para o PRPV como uma alternativa de desempenho robusto e ressaltaram que, em média, as soluções obtidas na CS estavam sempre próximas das melhores já encontradas na literatura. A CS encontrou boas soluções em tempo razoável, mantendo um desempenho computacional competitivo.

### 3 METODOLOGIA

A metodologia abordada neste trabalho consiste em aplicar a meta-heurística híbrida *Clustering Search* (CS) para resolução do Problema de Alocação de Equipes de Inventário Florestal (PAEIF). O modelo matemático utilizado neste trabalho é proposto por Barros Junior (2016) como uma adaptação ao Problema de Roteamento Periódico de Veículos com Janelas de Tempo (PRPVJT).

A meta-heurística híbrida CS foi modelada e implementada na linguagem C++. Os testes foram realizados em um computador com processador Intel Core I5 2.60GHz, 8GB de memória RAM. Para os mesmos, foram utilizadas instâncias definidas a partir de dados fornecidos por uma empresa de celulose sediada no estado do Espírito Santo. Cada instância contém o número talhões, o horizonte de planejamento (dias úteis do ano), uma matriz de distância na qual são considerados também os pontos de origem e destino, uma matriz binária de janela de tempo, e um vetor de tempo de medição de cada talhão.

#### 3.1 REPRESENTAÇÃO DE UMA SOLUÇÃO

A representação da solução adotada neste trabalho considera o número de dias, número de equipes e talhões visitados em determinado dia. A solução é representada por uma matriz de  $n$  linhas por  $m_k$  colunas, sendo  $n = \text{número de dias} * \text{número de equipes}$ , e  $m_k$  representa o número de talhões a serem visitados pela equipe  $e_{ij}$ . Cada linha da matriz solução de  $n$  linhas representa uma equipe  $e_{ij}$ , em que  $i$  representa o dia de trabalho e  $j$  o índice da equipe. Cada coluna representa um talhão a ser visitado por  $e_{ij}$ . As linhas possuem diferentes quantidades de talhões a serem visitados ( $m_k$ ), sendo que  $k$  representa o índice da linha em questão, e  $m_k$  pode assumir valores entre zero e o número máximo de talhões.

Considerando uma programação elaborada com 24 talhões à serem visitados, com 6 dias de trabalho e 2 equipes disponíveis, uma possível solução é apresentada na Figura 1. Nesse exemplo, cada linha da matriz representa uma rota, e por considerar duas equipes, cada dia de planejamento é representado por pares de linhas. As demais colunas representam os talhões a serem atendidos. Nesse exemplo, a equipe 1 no dia 2 deverá visitar os talhões 19 e 15 respectivamente.

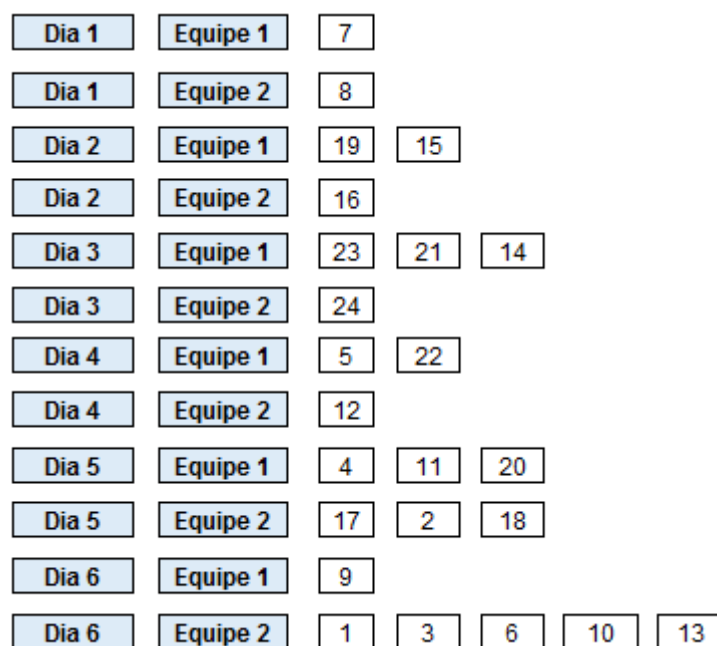


Figura 1 - Representação de uma de solução com 24 talhões e 6 dias.

A representação da quantidade de talhões visitados para cada equipe é feita através o vetor  $m_k$ . A Tabela 1 apresenta os valores presentes em  $m_k$  para a solução ilustrada na Figura 1.

Tabela 1 - Representação da Quantidade de Clientes.

| Linha $k$ da matriz de solução | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Quantidade de talhões          | 1 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 3 | 3  | 1  | 5  |

### 3.2 ESTRUTURAS DE VIZINHANÇA

Uma estrutura de vizinhança pode ser definida pelos movimentos entre os componentes da solução, ou seja, as diferentes estruturas de vizinhança são obtidas a partir de uma operação chamada movimento. Seja uma solução  $s$  conhecida, uma solução  $s'$  pode ser obtida por meio de três movimentos de troca realizados na matriz de solução.

Para este trabalho, foram desenvolvidos 3 tipos de movimentos, que são descritos a seguir:

### 3.2.1 Movimento Troca Sequência de Visita

O movimento Troca Sequência de Visita (TSV) realiza a troca entre dois talhões presentes na mesma linha da matriz solução. Esta troca altera a sequência dos talhões a serem visitados pela equipe. Na seguinte forma:

1. O método seleciona aleatoriamente uma linha cujo número de talhões seja superior à dois;
2. Uma vez selecionada a linha, seleciona-se dois talhões a serem atendidos pela mesma, também de forma aleatória;
3. O movimento realiza a troca entre os dois talhões escolhidos.

A Figura 2 apresenta um exemplo onde os talhões 6 e 13 são escolhidos para aplicação do movimento TSV. A matriz solução após o movimento de troca é exibida na Figura 3.

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

Figura 2 - Matriz Solução antes do movimento Troca Sequência de Visita.

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

Figura 3 - Matriz Solução depois do movimento Troca Sequência de Visita.

### 3.2.2 Troca Talhão de Equipe

O movimento Troca Talhão de Equipe (TTE) realiza a troca de equipe para um determinado talhão escolhido aleatoriamente. Por manter um talhão no mesmo dia de atendimento, ao receber uma solução  $s$ , que respeite a restrição de janela de tempo, a solução  $s'$  gerada por este movimento, também não infringe a restrição de janela de tempo. O movimento possui os seguintes passos:

1. O método seleciona aleatoriamente um talhão qualquer da matriz solução;
2. Uma vez selecionado o talhão, uma nova equipe para este é definida, também de forma aleatória;
3. O talhão escolhido no passo 1 é alocado à última posição da equipe escolhida no passo 2.

Considere que o talhão 14 seja escolhido ao acaso. A Figura 4 apresenta uma solução na qual o talhão é atendido pela equipe 1 no dia 3. A Figura 5 exhibe a matriz solução após a realização do movimento TTE. Nesse exemplo a equipe 2 é selecionada para atender o talhão. Observe que o talhão 14 passa a ser visitado pela equipe 2 no mesmo dia 3.



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

Figura 4 - Matriz Solução antes do movimento Troca Talhão de Equipe.

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

Figura 5 - Matriz Solução antes do movimento Troca Talhão de Equipe.

### 3.2.3 Troca Talhão de Dia

O movimento Troca Talhão de Dia (TTD) realiza a troca de dia para um talhão escolhido aleatoriamente, desde que o novo dia seja viável para este ser alocado. Dessa forma, ao receber uma solução  $s$ , que respeite a restrição de janela de tempo,

a solução  $s'$  gerada por este movimento também não violará a restrição de janela de tempo. O movimento possui os seguintes passos:

1. O método seleciona aleatoriamente um talhão qualquer da matriz solução;
2. Uma vez escolhido o talhão, verifica-se a existência de um dia alternativo ao atual para que este seja atendido;
3. Uma equipe é selecionada aleatoriamente para atender o talhão no dia definido no passo 2;
4. O talhão é alocado na última posição da equipe no dia definido.

Considere que o talhão 15 é escolhido aleatoriamente. A Figura 6 apresenta uma solução na qual este talhão é atendido no dia 2 pela equipe 1, dentro de sua janela de tempo. A Figura 7 exibe uma nova solução na qual o talhão 15 passa a ser atendido pela equipe 2 no dia 1.

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

Figura 6 - Matriz Solução antes do movimento Troca Talhão de Dia.

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

Figura 7 - Matriz Solução depois do movimento Troca Talhão de Dia.

A Figura 8 representa a janela de tempo para o talhão 15, especificando os dias viáveis como aqueles em que o talhão 15 poderá ser atendido e inviáveis os dias fora da sua janela de tempo de atendimento.

| Dia             | 1      | 2      | 3      | 4        | 5        | 6        |
|-----------------|--------|--------|--------|----------|----------|----------|
| Janela de Tempo | Viável | Viável | Viável | Inviável | Inviável | Inviável |

Figura 8 - Exemplo de matriz de janela de tempo para o talhão 15.

### 3.3 CLUSTERING SEARCH

A *Clustering Search* (CS) é uma meta-heurística híbrida baseada no algoritmo *Evolutionary Clustering Search* (ECS) descrito por Oliveira e Lorena (2004). Segundo Oliveira et al. (2013) a principal adaptação proposta na CS ocorre pela variação do algoritmo evolucionário, como gerador de solução, por meta-heurísticas em geral. O funcionamento da CS se dá pela concepção de um conjunto de soluções através de uma meta-heurística e aplicação de uma busca local nas regiões consideradas promissoras a terem boas soluções.

A escolha da CS para este trabalho está relacionada aos bons resultados encontrados por Araújo et al. (2013), conforme descrito na Seção 2.3, e pela pouca variedade de trabalhos encontrados na literatura que utilizam a CS para os problema de roteamento de veículos. Por ser considerada uma meta-heurística híbrida, a CS possibilita a

variação do método responsável por gerar soluções, combinando assim diferentes métodos. Nesse trabalho, o *Simulated Annealing* (SA) é utilizado como meta-heurística geradora de solução.

Oliveira et al. (2013) descrevem a CS como um método que realiza agrupamentos de soluções em *clusters* para identificação de regiões possivelmente promissoras. Um *cluster*  $C_i$  é composto por três atributos, a saber,  $C = \{c, v, r\}$  sendo que  $c_i$  representa o centro do *cluster*,  $v_i$  o volume do cluster e  $r_i$  o seu índice de ineficiência. O volume  $v_i$  representa a quantidade de soluções agrupadas ao *cluster*  $C_i$ . Quando  $v_i$  alcançar um valor máximo  $\lambda$ , previamente definido, a região poderá ser interpretada como promissora. Quando uma região é considerada promissora, verifica-se o índice de ineficiência  $r_i$ . Tal índice analisa a eficiência da busca local aplicada ao *cluster*  $C_i$ . A cada iteração de uma região promissora sem melhora com a busca local, o índice de ineficiência  $r_i$  é acrescido em uma unidade. Ao atingir um valor  $r_{max}$ , o índice  $r_i$  indica a necessidade de uma perturbação ao centro  $c_i$ . Dessa forma, é possível proporcionar variabilidade à solução, permitindo que a CS escape de regiões demasiadamente exploradas. A Figura 9, apresenta o fluxograma de execução da CS proposto por Chaves e Lorena (2010).

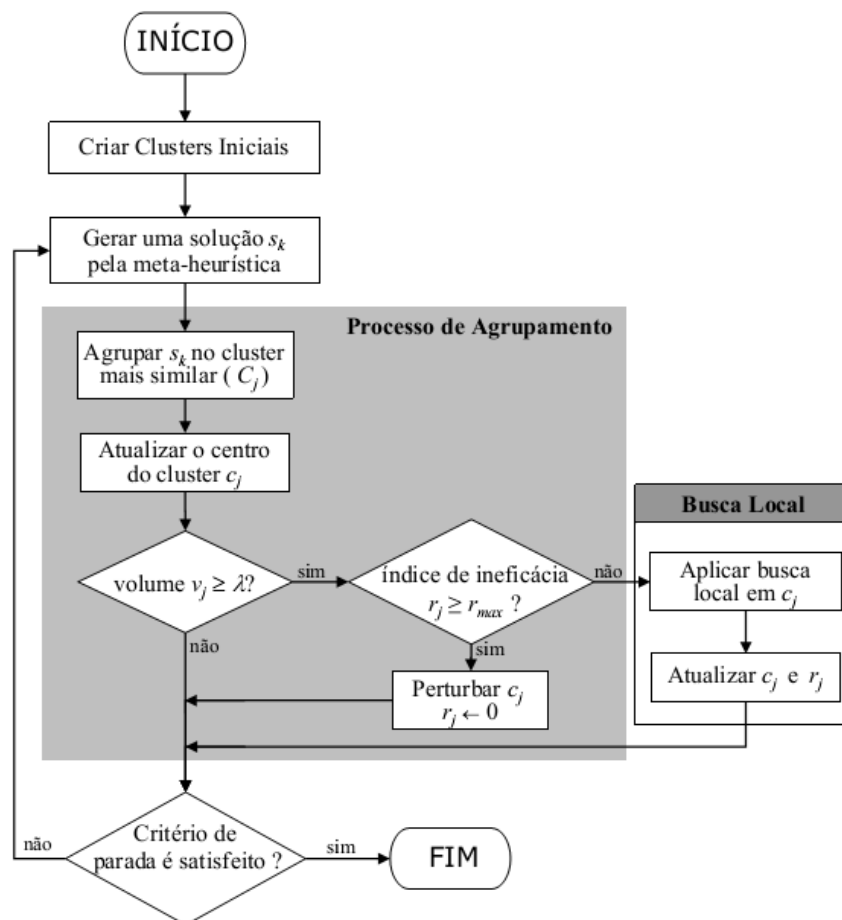


Figura 9 - Fluxograma da CS.  
Fonte: Chaves e Lorena (2010).

Ainda completam a CS uma meta-heurística geradora de solução, um processo de agrupamento e uma heurística de busca local.

### 3.3.1 Geradora de Solução

A meta-heurística *Simulated Annealing* (SA), proposta por Kirkpatrick et al. (1983), é um método de busca local no qual é permitido realizar movimentos que gerem soluções piores que as atuais a fim de fugir de possíveis ótimos locais.

Uma solução  $s$ , já conhecida, é dada ao algoritmo que gera uma solução  $s'$  vizinha de  $s$ . A cada geração de  $s'$  é verificada a variação da função objetivo  $\Delta$ , sendo  $\Delta = f(s') - f(s)$ . Se  $\Delta < 0$ , a solução vizinha  $s'$  possui uma função objetivo melhor que  $s$  (em problemas de minimização), portanto  $s'$  se torna a solução  $s$ . Caso  $\Delta \geq 0$  a probabilidade de aceitação  $e^{(-\frac{\Delta}{T})}$  é calculada, sendo  $T$  a temperatura com um valor

previamente definido. A probabilidade de aceitação ajusta a aceitação ou não de soluções piores que a atual. Quando a temperatura se aproxima de zero, ou seja, quando soluções piores que a atual passam a não ser mais aceitas, o método é encerrado. O pseudocódigo da meta-heurística SA é apresentado na Figura 10.

```

1. Dado ( $\alpha$ ,  $SA_{max}$ ,  $T_0$ ,  $T_c$  e  $s$ )
2.  $s^* \leftarrow s$            {Melhor solução obtida até então}
3.  $iter \leftarrow 0$        {Número de iterações na temperatura  $T$ }
4.  $T \leftarrow T_0$        {Temperatura corrente}
5. ENQUANTO  $T > T_c$  FAÇA
6.   ENQUANTO  $iter < SA_{max}$  FAÇA
7.      $iter \leftarrow iter + 1$ 
8.      $s' \leftarrow N(s)$ 
9.      $\Delta \leftarrow f(s') - f(s)$ 
10.    SE  $\Delta < 0$  ENTÃO
11.       $s \leftarrow s'$ 
12.    SE  $(f(s') < f(s^*))$  ENTÃO
13.       $s^* \leftarrow s'$ 
14.    FIM-SE;
15.    SENÃO
16.      considere  $x \in [0,1]$ 
17.      SE  $(x < e^{-\Delta/T})$  ENTÃO
18.         $s \leftarrow s'$ 
19.      FIM-SE
20.    FIM-SE
21.  FIM-ENQUANTO
22.   $T \leftarrow \alpha T$ 
23.   $iter \leftarrow 0$ 
24. FIM-ENQUANTO
25. Retornar  $s^*$ 

```

Figura 10 - Pseudocódigo da SA  
Adaptado de Mauri (2005).

### 3.3.2 Agrupamento Iterativo

O agrupamento iterativo é responsável por alocar as soluções geradas pela meta-heurística aos *clusters* mais similares a estas. A análise de similaridade entre uma solução  $s$  e um *cluster*  $C_i$  é feita através do cálculo da distância de Hamming  $d_{Ham}$  (HAMMING, 1950). Esse cálculo é feito por meio da contagem do número de desigualdades entre a solução  $s$  e o centro do *cluster*  $C_i$ .

A Figura 11 proposta por Chaves (2009) apresenta duas soluções e a realização do cálculo, os quadrados destacados em cinza mostram os talhões distintos entre as duas soluções. Ao final da etapa de agrupamento, a solução  $s$  será alocada ao *cluster*  $C_i$  menos distante, ou seja, ao *cluster* em que obtiver menor  $d_{Ham}$ .

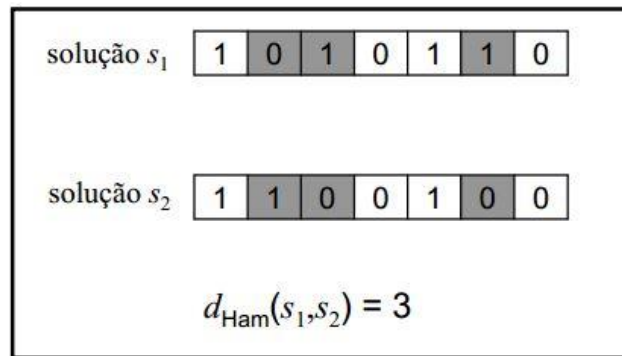


Figura 11 - Exemplo do cálculo da distância de Hamming.

Caso a solução  $s$  seja melhor, quando comparada a  $c_i$ , centro do *cluster*  $C_i$ ,  $s$  passará a ser o novo centro do *cluster*. Caso  $s$  não seja melhor que  $c_i$  ela será somente agrupada ao *cluster*  $C_i$ .

Ao agrupar uma solução  $s$  em um *cluster*  $C_i$ , uma análise é realizada no *cluster* verificando se o seu volume atingiu o limite máximo  $\lambda$ , ou seja, se este *cluster* é uma região promissora para realização de uma busca local. O índice de ineficiência  $r_i$  é verificado para saber se este atingiu  $r_{max}$ . Caso  $r_i < r_{max}$  uma busca local é realizada ao centro do *cluster*  $C_i$ . Se  $r_i > r_{max}$  uma perturbação é realizada ao centro do *cluster*, simbolizando que uma busca local foi aplicada mais de  $r_{max}$  vezes em  $C_i$  sem obter melhora.

### 3.3.3 Busca local

Como busca local foi desenvolvida um heurística de busca local para alternar os talhões explorando possíveis melhoras na solução. Nessa busca, todos os talhões a serem atendidos são alocados em uma nova linha na matriz de solução, podendo ser atribuído a uma equipe em um novo dia, ou a outra equipe no mesmo dia. A definição do novo dia em que o talhão será alocado é condicionada ao atendimento da restrição de janela de tempo. Após a remoção de um talhão em determinada linha, são realizados nesta linha movimentos que alternem a sequência de visitas, a fim de verificar uma nova sequência de visitas melhor que a atual.

Definida a nova linha na qual o talhão será alocado, verifica-se para todas as posições da linha, em qual ordem o talhão poderá ser atendido proporcionando melhor ganho na FO. Se a busca resultar em uma solução melhor que a atual, esta passará a ocupar o posto de melhor solução e a busca local realizará novas buscas a partir da melhor

solução. Caso a busca apresente piora em relação a melhor solução, esta será desfeita e a busca local continua considerando a melhor solução anterior.

### 3.3.4 Função objetivo

A função objetivo visa minimizar o tempo de deslocamento entre os talhões. Neste trabalho, são adotadas técnicas de penalização na função objetivo sempre que uma solução se mostrar inviável, ou seja, sempre que uma solução violar alguma restrição. Utiliza-se tal estratégia a fim não permitir que soluções inviáveis sejam aceitas. Seja  $f(s)$  a função objetivo da solução  $s$ , um valor de penalização  $p_i$  é acrescentado a  $f(s)$  sempre que este violar uma restrição  $r_i$  (em problemas de minimização), sendo cada penalização  $p_i$  referente a uma restrição  $r_i$ .

As penalizações  $p_i$  aplicáveis a  $f(s)$  são definidas por  $p_1$ ,  $p_2$  e  $p_3$ . A penalização  $p_1$  é aplicada sempre que algum talhão presente na solução  $s$  for atendido fora de sua janela de tempo previamente estabelecida. Sempre que uma equipe não realizar atendimento em determinado dia, ou quando o tempo de deslocamento somado ao tempo de serviço da equipe for superior as oito horas diárias, atribuem-se a  $f(s)$  as penalizações  $p_2$  e  $p_3$ , respectivamente. A função objetivo para o problema é dada por:

$$f(s) = \sum_{i \in T} \sum_{k \in K} \sum_{i \in \{L_r \cup Go\}} \sum_{j \in \{L_r \cup Gd\}} d_{i,j} x_{i,j,k}^t \quad \forall i \neq j; i = Go \mid j \neq Gd; j = Gd \mid i \neq Go$$



### 3.3.5 Pseudocódigo da CS

A Figura 12 apresenta o pseudocódigo para a *Clustering Search* (CS).

```

1. Criar  $\gamma$  novas soluções (clusters) aleatoriamente
2.  $v_i \leftarrow 1$  e  $r_i \leftarrow 0 \forall i = 1, \dots, \gamma$ 
3.  $s \leftarrow$  solução inicial
4.  $s^* \leftarrow s$   $T \leftarrow T_0$ 
5. ENQUANTO  $T > T_c$  FAÇA
6.    $iter \leftarrow 0$ 
7.   ENQUANTO  $iter < SA_{max}$  FAÇA
8.      $iter \leftarrow iter + 1$ 
9.      $s' \leftarrow N(s)$ 
10.     $\Delta \leftarrow f(s') - f(s)$ 
11.    SE  $\Delta < 0$  ENTÃO
12.       $s \leftarrow s'$ 
13.      SE  $f(s') > f(s)$  ENTÃO
14.         $s \leftarrow s'$ 
15.      FIM-SE
16.      SENÃO
17.        Com probabilidade  $e^{-(f(s)-f(s'))/T}$   $s \leftarrow s'$ 
18.        FIM-SE
19.      FIM-ENQUANTO
20.     $T \leftarrow \alpha T$ 
21.     $i \leftarrow \arg \min_{i \in \{1, \dots, \gamma\}} \{H_i\}$ 
22.     $v_i \leftarrow v_i + 1$ 
23.     $c_i \leftarrow$  melhor( $s, c_i$ )
24.    SE  $v_i = \lambda$  ENTÃO
25.       $v_i \leftarrow 0$ 
26.       $s \leftarrow$  busca local( $c_i$ )
27.      SE  $f(s) = f(c_i)$  ENTÃO
28.         $r_i \leftarrow r_i + 1$ 
29.        SE  $r_i = r_{max}$  ENTÃO
30.           $r_i \leftarrow 0$ 
31.           $c_i \leftarrow N(c_i)$ 
32.          FIM-SE
33.        SENÃO
34.           $r_i \leftarrow 0$ 
35.           $c_i \leftarrow s$ 
36.          FIM-SE
37.        FIM-SE
38.       $s^* \leftarrow \max(s^*, c_i)$ 
39.    FIM-ENQUANTO
Retornar  $s^*$ 

```

Figura 12 - Pseudocódigo da CS utilizando SA.

## 4 RESULTADOS

Os valores atribuídos as penalizações  $p_1$  e  $p_2$  foram definidos empiricamente. A penalização  $p_3$  é definida pelo *tempo de deslocamento + tempo de serviço*. O cálculo utilizado para definição de  $p_3$  atribui um valor proporcional ao tempo excedente de trabalho à  $f(s)$ . Os valores para  $p_1$  e  $p_2$  foram encontrados após uma determinada sequência de testes serem realizados. Por se tratar de um problema de minimização, esses valores são propositalmente elevados, evitando que soluções inviáveis sejam aceita. A Tabela 2 apresenta os valores para as penalizações.

Tabela 2 - Valores das penalizações.

| Penalização | Valor |
|-------------|-------|
| $p_1$       | 1000  |
| $p_2$       | 1000  |

Durante os testes realizados, foram verificadas diversas combinações entre os parâmetros do SA e da CS. Os valores foram definidos empiricamente e ajustados de acordo com os resultados obtidos nos testes. A combinação de parâmetros e seus respectivos valores estão dispostos na Tabela 3.

Tabela 3 – Combinação de valores para os parâmetros.

|    |            |      |      |       |
|----|------------|------|------|-------|
| SA | $SA_{max}$ | 1000 | 5000 | 10000 |
|    | $\alpha$   | 0,95 | 0,95 | 0,975 |
|    | $T_0$      | 100  | 1000 | 1000  |
|    | $T_C$      | 0,01 | 0,01 | 0,01  |
| CS | $\gamma$   | 7    | 10   | 10    |
|    | $\lambda$  | 5    | 5    | 7     |
|    | $r_{max}$  | 5    | 3    | 3     |

As combinações realizadas com os parâmetros da SA são:  $SA_{max}$  que representa a quantidade de iterações do SA (1000, 5000, 10000),  $\alpha$  que indica a taxa de resfriamento (0,95, 0,95, 0,975),  $T_0$  é a temperatura inicial (100, 1000, 1000), e  $T_C$  é temperatura de congelamento (0,01). Os parâmetros da CS são:  $\gamma$  indica a quantidade de *clusters* (7, 10, 10),  $\lambda$  é o volume máximo para cada *cluster* (5, 5, 7),  $r_{max}$  é o índice de ineficácia máximo para cada *cluster* (5, 3, 3). Foram realizados testes utilizando as

mesmas instâncias, sem limite de tempo, com critério de parada sendo  $SA_{max}$ . Ao final dos testes, a combinação de parâmetros que obteve melhores resultados é apresentada na Tabela 4.

Tabela 4 - Valores dos parâmetros.

|    |            |       |
|----|------------|-------|
| SA | $SA_{max}$ | 10000 |
|    | $\alpha$   | 0,975 |
|    | $T_0$      | 1000  |
|    | $T_c$      | 0,01  |
| CS | $\gamma$   | 10    |
|    | $\lambda$  | 7     |
|    | $r_{max}$  | 5     |

Conhecida a combinação de parâmetros apresentada na Tabela 4, foram executados testes a fim de avaliar o desempenho da meta-heurística e analisar qual dos movimentos de perturbação proporcionam melhores soluções. A cada instância de entrada fornecida, a CS foi executada 10 vezes para determinado movimento de perturbação. Também foi testada uma alternativa que utiliza todos os movimentos, definindo aleatoriamente qual utilizar a cada perturbação. A Tabela 5 apresenta os resultados obtidos na realização dos testes com um comparativo entre os movimentos. A instância de 1000 talhões considera um cenário com 3 equipes de trabalho. Já as demais instâncias consideram um cenário com 2 equipes de trabalho.

Tabela 5 - Resultados de dez execuções do CS para cada movimento.

| Movimento                                | Instância | Melhor FO        | Média     | Desvio | Tempo FO (seg) | Viável % |
|--|-----------|------------------|-----------|--------|----------------|----------|
| Troca<br>sequência<br>de visita<br>(TSV) | 24p_6     | 140,52           | 146,15    | 4,01   | 4,00           | 40       |
|  | 50p_12    | 2.496,91         | 2.598,60  | 4,07   | 112,00         | 50       |
|  | 100p_24   | 4.363,81         | 4.389,75  | 0,59   | 85,00          | 20       |
|  | 100p_36   | -                | -         | -      | -              | 0        |
|  | 200p_48   | 8.953,96         | 9.074,98  | 1,35   | 148,00         | 20       |
|  | 400p_60   | -                | -         | -      | -              | 0        |
|  | 500p_60   | -                | -         | -      | -              | 0        |
|  | 1000p_120 | -                | -         | -      | -              | 0        |
| Troca<br>Talhão de<br>equipe<br>(TTE)    | 24p_6     | 140,02           | 144,87    | 3,46   | 8,00           | 80       |
|  | 50p_12    | 2.472,81         | 2.662,63  | 7,68   | 73,00          | 50       |
|  | 100p_24   | 4.261,75         | 4.284,15  | 0,53   | 58,00          | 20       |
|  | 100p_36   | -                | -         | -      | -              | 0        |
|  | 200p_48   | -                | -         | -      | -              | 0        |
|  | 400p_60   | -                | -         | -      | -              | 0        |
|  | 500p_60   | -                | -         | -      | -              | 0        |
|  | 1000p_120 | -                | -         | -      | -              | 0        |
| Troca<br>Talhão de<br>dia (TTD)          | 24p_6     | 138,62           | 141,52    | 2,09   | 93,00          | 100      |
|  | 50p_12    | 2.398,62         | 2.565,29  | 6,95   | 121,00         | 80       |
|  | 100p_24   | 4.021,62         | 4.067,59  | 1,14   | 146,00         | 100      |
|  | 100p_36   | 6.106,17         | 6.138,03  | 0,52   | 145,00         | 100      |
|  | 200p_48   | 7.866,70         | 7.925,05  | 0,74   | 154,00         | 100      |
|  | 400p_60   | 11.549,62        | 11.784,16 | 2,03   | 135,00         | 80       |
|  | 500p_60   | -                | -         | -      | -              | 0        |
|  | 1000p_120 | -                | -         | -      | -              | 0        |
| Híbrido                                  | 24p_6     | <b>132,61</b>    | 133,42    | 0,61   | 140,00         | 100      |
|  | 50p_12    | <b>2.322,70</b>  | 2.348,94  | 1,13   | 143,00         | 100      |
|  | 100p_24   | <b>4.007,15</b>  | 4.037,21  | 0,75   | 161,00         | 100      |
|  | 100p_36   | <b>6.097,26</b>  | 6.116,08  | 0,31   | 159,00         | 100      |
|  | 200p_48   | <b>7.796,29</b>  | 7.866,81  | 0,90   | 165,00         | 100      |
|  | 400p_60   | <b>11.290,15</b> | 11.465,88 | 1,56   | 151,00         | 100      |
|  | 500p_60   | <b>11.716,64</b> | 11.834,99 | 1,01   | 154,00         | 100      |
|  | 1000p_120 | <b>34.711,63</b> | 35.412,61 | 2,01   | 5.427,00       | 100      |

A Tabela 5 apresenta os resultados obtidos agrupados por tipo de movimento. A coluna movimento identifica qual movimento de perturbação foi utilizado, a coluna

instância abrevia qual entrada está sendo utilizada para leitura dos dados. Nessa coluna adota-se um padrão de nomeação para as instâncias de entrada numeroDeTalhõesp\_quantidadeDeDias. Considere a instância de 24 talhões e 6 dias sua sigla será 24p\_6. A coluna melhor FO indica qual a melhor função objetivo obtida para as dez execuções de determinado movimento. A coluna Média indica uma média entre as FO's de soluções viáveis obtidas nas dez execuções. A coluna desvio padrão indica o desvio padrão entre as soluções viáveis obtidas. A coluna Tempo FO indica o tempo em segundos no qual foi encontrada a melhor FO. A coluna Viável % indica qual a porcentagem de soluções viáveis obtidas entre o total de execuções para cada movimento com determinada instância.

De acordo com os resultados apresentados na Tabela 5, pode-se observar que os movimentos TSV e TPE, quando utilizados em conjunto com a CS, não conseguem encontrar soluções viáveis para todas as instâncias. O movimento TTD se mostrou mais eficiente em relação aos demais, conseguindo encontrar soluções viáveis para instancias de até 400 talhões e 60 dias. Entretanto, quando utilizados de forma conjunta, os três movimentos apresentam uma melhora significativa, como pôde ser observado nos resultados descritos na seção “Híbrido” da Tabela 5.

A Tabela 6 apresenta um comparativo da CS com as soluções obtidas pelo *software* CPLEX.

Tabela 6 – Comparação com as soluções obtidas pelo CPLEX.

| Instância | Clustering Search |                  |                 | CPLEX           |          |              |         |
|-----------|-------------------|------------------|-----------------|-----------------|----------|--------------|---------|
|           | FO                | Media            | Tempo (seg.)    | FO              | Lim.     | Tempo        | Gap (%) |
| 24p_6     | <b>132,61</b>     | 133,42           | 140,00          | <b>132,61</b>   | 132,03   | <b>43,78</b> | 0,43    |
| 50p_12    | 2.322,70          | 2.348,94         | 143,00          | <b>2.318,97</b> | 2.190,26 | 1.785,45     | 5,55    |
| 100p_24   | <b>4.007,15</b>   | <b>4.037,21</b>  | <b>161,00</b>   | -               | 3.722,79 | 357,22       | -       |
| 100p_36   | <b>6.097,26</b>   | <b>6.116,08</b>  | <b>159,00</b>   | 6.446,30        | 5.951,27 | 451,67       | 7,68    |
| 200p_48   | <b>7.796,29</b>   | <b>7.866,81</b>  | <b>165,00</b>   | -               | 6.307,32 | 9.202,89     | -       |
| 400p_60   | <b>11.290,15</b>  | <b>11.465,88</b> | <b>151,00</b>   | -               | -        | -            | -       |
| 500p_60   | <b>11.716,64</b>  | <b>11.834,99</b> | <b>154,00</b>   | -               | -        | -            | -       |
| 1000p_120 | <b>34.711,63</b>  | <b>35.412,61</b> | <b>5.427,00</b> | -               | -        | -            | -       |

A Tabela 6 dispõe as soluções encontradas pela CS em comparação ao CPLEX. A coluna instância abrevia qual entrada está sendo utilizada para leitura dos dados. Em relação as informações da CS na tabela, a coluna FO indica a melhor FO obtida, a coluna media indica o valor médio entre as FO's obtidas, e a coluna tempo informa o instante de tempo em segundos no qual a melhor FO foi obtida. Em relação as informações do CPLEX na tabela, a coluna FO indica a melhor FO obtida, a coluna Limitante indica o melhor valor conhecido para a FO, a coluna tempo indica o tempo de execução e a coluna GAP indica a distância do valor da função objetivo da melhor solução obtida pelo CPLEX para a FO da solução ótima.

Para todas as instâncias testadas, a CS não obteve resultados superiores ao *software* CPLEX apenas para a instância de 50 talhões e 12 dias (50p\_12). Para as demais instâncias, a CS encontra resultados melhores e também resultados não encontradas pelo CPLEX. Na instância 100p\_36, a CS obteve uma solução 5,41% melhor que a solução encontrada pelo CPLEX. Para a instancia 50p\_12, a CS conseguiu resultados próximos à solução do CPLEX, porém não conseguiu melhorar a solução e manteve-se 0,16% acima. Na instância 24p\_6 a CS chegou ao mesmo resultado que o método exato, porém em um tempo superior.

A CS obteve soluções que permitiram reduzir a quantidade de dias para a instancia de 100 talhões, como pode ser visto na Tabela 6. O CPLEX encontrou solução apenas para um cenário com 100 talhões em 36 dias. Já a CS encontra soluções viáveis para uma instância com 100 talhões em 24 dias, possibilitando a redução de 12 dias no horizonte de planejamento. Para as instâncias com 400, 500 e 1000 talhões, a CS obteve resultados com soluções viáveis, porém o CPLEX não encontrou nenhum valor para parâmetro comparativo.

## 5 CONCLUSÕES

O presente trabalho abordou o Problema de Alocação de Equipes de Inventário Florestal (PAEIF) apresentado por Barros Junior (2016). O modelo utilizado para este problema é fundado no modelo proposto para o Problema de Roteamento Periódico de Veículos com Janelas de Tempo (PRPVJT).

Para a resolução do problema proposto foi utilizada a meta-heurística *Clustering Search* usando a *Simulated Annealing* como geradora de soluções

Os resultados obtidos pela CS para o PPIF foram satisfatórios e, em alguns casos, superiores às soluções obtidas pelo CPLEX. Na melhor ocorrência, entre as instâncias em que o CPLEX encontrou resultados, a CS foi capaz de melhorar em 5,41% a melhor solução encontrada pelo método exato. A CS obteve soluções viáveis com um desvio baixo para todas as instâncias propostas neste trabalho e encontrou soluções que possibilitam até mesmo reduzir o horizonte de planejamento de 36 dias para 24 dias, para a instância de 100 talhões.

A CS se mostrou como uma boa alternativa para a resolução do PAEIF, utilizando um modelo baseado no PRPVJT. Em trabalhos futuros, novas instâncias para o problema poderão ser exploradas, e uma nova meta-heurística geradora de solução pode ser combinada a CS.

## 6 REFERÊNCIAS

ARAÚJO, E.; POLDI, K.; CHAVES, A. Método híbrido clustering search aplicado ao problema de roteamento periódico. In: **Simpósio de aplicações operacionais em áreas de defesa**, 15. 2013, São José dos Campos.

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa operacional**. 6 ed. Rio de Janeiro, RJ: Elsevier, 2007. 524 p.

BARROS JUNIOR. A. A.; **Aplicação de meta-heurísticas no planejamento do inventário florestal**. 2016. Tese de Doutorado - Centro de Ciências Agrárias, Universidade Federal do Espírito Santo, Alegre. (Dados de pesquisa ainda não publicados).

CARVALHO, M. H.; CERIOLI, M.R.; DAHAB, R.; FEOFILOFF, P.; FERNANDES, C. G.; FERREIRA, C. E.; GUIMARÃES, K. S.; MIYAZAWA, F. K.; PINA JR, J. C.; SOARES, J. A. R.; WAKABAYASHI, Y. **Uma introdução sucinta a algoritmos de aproximação**. 23º Colóquio Brasileiro de Matemática: [s.n.], p. 157, 2001.

CAMPOS, J. C. C.; LEITE, H. G. **Mensuração florestal: perguntas e respostas**. Viçosa: UFV, 2006. 470 p.

CHAVES, A. A. **Uma meta-heurística híbrida com busca por agrupamentos aplicada a problemas de otimização combinatória**. 2009. 199f. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.

CHAVES, A. A.; LORENA, L. A. N. Clustering search algorithm for the capacitated centered clustering problem. **Computers & Operations Research**, v. 37, p. 552-558, 2010.

CORDEAU J. F; LAPORTE, G.; MERCIER, A. A. Unified tabu search heuristic for vehicle routing problems with time windows. **The Journal of the Operational Research Society**. v. 52, no. 8, p. 928-93, Agosto. 2011.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, 109-133, 1995.



GOLDEN, B. L.; CHAO, I. M.; WASIL, E. (1995), An Improved Heuristic for the Periodic Vehicle Routing Problem. **Networks**, 25-44, 1995.

HAMMING, R. W. Error detecting and error correcting codes. **Bell System Technical Journal**, v. 26, p. 147-160, 1950.

HUSCH, B.; MILLER, C.I; BEERS, T.W. Forest mensuration. 3.ed. Malabar: **Krieger Publishing Company**, 1993. 402 p.

KIRKPATRICK, S.; GELATT, C.; VECCHI, M. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 498-516, 1983.

LOESCH, C.; HEIN, N. Pesquisa Operacional: Fundamentos e modelos. **Saraiva**, São Paulo, 2009.

MAURI, G. R. **Novas heurísticas para o problema de escalonamento de tripulações**. Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisa Espacial, São José dos Campos, 2005.

MENEGUZZI, C. C. **Modelo de roteamento de veículos aplicado ao planejamento do inventário florestal**. 2011, 94 f. Dissertação (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.; TOULOUSE, M. A Hybrid Genetic Algorithm for the Periodic Vehicle Routing Problem with Time Windows. **CIRRELT**, Montréal, Canada, v. 25, abr. 2011.

OLIVEIRA, A. C. M; LORENA, L. A. N. Detecting promising areas by evolutionary clustering search. **Lecture Notes in Artificial Intelligence**, v. 3171, p. 385-394, 2004.

OLIVEIRA, A, C. M.; CHAVES, A. A; LORENA, L. A N. Clustering Search. **Pesquisa Operacional**. v. 33, n. 1, p.105-121, 2013.

PILLAC V.; GENDREAU, M.; GUÉRET, C.; MEDAGUIA, A. L. A review of dynamic vehicle routing problems. **European Journal of Operational Research**. v. 225, p. 1-11, 2013.

SOARES, C. P. B.; NETO, F. P.; SOUZA, A. L. **Dendrometria e inventário florestal**. 1 ed. Viçosa, MG: UFV, 2007. 276 p.

