UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE COMPUTAÇÃO

ROBSON DE SOUZA MELO

VARIAÇÕES DA META-HEURÍSTICA *CLUSTERING SEARCH* APLICADAS NA RESOLUÇÃO DE UM PROBLEMA DE PLANEJAMENTO FLORESTAL

ALEGRE - ES
2014
ROBSON DE SOUZA MELO

VARIAÇÕES DA META-HEURÍSTICA *CLUSTERING SEARCH* APLICADAS NA RESOLUÇÃO DE UM PROBLEMA DE PLANEJAMENTO FLORESTAL

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

Orientador: Prof. Dr. Geraldo Regis Mauri.

Co-Orientador: Prof. MSc. Antonio Almeida Barros Junior.

ALEGRE - ES

2014

ROBSON DE SOUZA MELO

VARIAÇÕES DA META-HEURÍSTICA *CLUSTERING SEARCH* APLICADAS NA RESOLUÇÃO DE UM PROBLEMA DE PLANEJAMENTO FLORESTAL

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

Aprovado em 03 de Dezembro de 2014.

BANCA EXAMINADORA

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

Prof. MSc. Antonio Almeida Barros Junior
Universidade Federal do Espírito Santo
Co-Orientador

Prof. MSc. Edmar Hell Kampke
Universidade Federal do Espírito Santo

“É necessário sempre acreditar que o sonho é possível, que o céu é o limite e que você truta é imbatível.”

Racionais MC’s.

AGRADECIMENTOS

A Deus, por me guiar e fazer com que meus caminhos sempre sigam para o sucesso.

A meus pais, pela ajuda financeira e apoio moral.

À minha esposa, por me manter focado nos meus objetivos.

Aos meus orientadores, por auxiliarem nos trabalhos e pelas dicas e conselhos recebidos.

Aos familiares, que sempre torceram muito por mim.

Aos amigos, por me ajudarem durantes estes anos difíceis.

RESUMO

Este trabalho propõe a utilização das meta-heurísticas *Iterated Local Search* (ILS) *Greedy Randomized Adaptive Search Procedure* (GRASP), *Simulated Annealing* (SA) e Algoritmo Genético (AG) como geradoras de soluções para a meta-heurísticas híbrida *Clustering Search* (CS). Além disso, a aplicação da técnica de intensificação *Path Relinking* (PR) também foi considerada, com intuito de obter melhores resultados. As meta-heurísticas foram aplicadas na resolução de um Problema de Planejamento Florestal (PPF). O PPF consiste na seleção de alternativas de manejo para plantio e colheita de madeira de forma a maximizar um valor associado ao plano de corte de unidades florestais dentro de um horizonte de planejamento. Portanto neste trabalho, é proposta a utilização da ILS, GRASP, SA e AG na fase de geração de soluções da CS, e a aplicação da técnica de intensificação PR para tentar intensificar os resultados. O desempenho dos métodos propostos foi avaliado por meio de experimentos computacionais realizados e comparados com diferentes métodos apresentados em trabalhos anteriores encontrados na literatura.

Palavras-chave: *Clustering Search*; Problema de Planejamento Florestal; Meta-heurísticas.

DIFFERENT CLUSTERING SEARCH APPROACHES TO SOLVE A FOREST PLANNING PROBLEM

ABSTRACT

This work proposes the use of metaheuristics Iterated Local Search (ILS) Greedy Randomized Adaptive Search Procedure (GRASP), Simulated Annealing (SA) and Genetic Algorithm (GA) to generate solutions for the hybrid metaheuristic known as Clustering Search (CS). A Path Relinking (PR) intensification technique was also employed looking for better results. The metaheuristics were applied to solve a Forest Planning Problem (FPP). The FPP consists of selecting management alternatives for planting and harvesting of timber in order to maximize a value associated with the cutting of forest units within a planning horizon. So in this work, we propose the use of ILS, GRASP, SA and GA for generating solutions for CS and the use of the PR technique to enhance the results. The performance of the proposed methods was evaluated through computational experiments and the results were compared with different methods reported in previous works found in the literature.

Keywords: Clustering Search; Forest Planning problem; Metaheuristics.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1INTRODUÇÃO.....10
6REVISÃO DA LITERATURA.....11
7METODOLOGIA.....12
12RESULTADOS.....19
13CONCLUSÕES.....32
14REFERÊNCIAS.....32

1 INTRODUÇÃO

O Problema de Planejamento Florestal (PPF) surgiu da necessidade de se ordenar a produção madeireira de forma a obter uma maior lucratividade. Essa não é uma tarefa trivial, pois o problema contém inúmeras restrições e alternativas de manejo. Com o crescimento das florestas e da produção madeireira, cresce também a necessidade de se planejar melhor o manejo das florestas. Para auxiliar o tomador de decisões, várias técnicas gerenciais foram desenvolvidas, dentre as quais se sobressaem as técnicas de Pesquisa Operacional (PO) (SILVA et al., 2009).

Algumas técnicas de PO vêm sendo largamente utilizadas na resolução de problemas combinatoriais e, dentre elas, se destaca a Programação Linear (PL). Porém, Silva (2001) lista algumas dificuldades com a PL como, por exemplo, a não integralidade das soluções, condição que, muitas vezes, leva à necessidade de subdividir um talhão, que é uma parte da floresta na qual as árvores contêm as mesmas características. Na prática, deve-se evitar que um talhão seja por subdividido.

Para Rodrigues (2001), a necessidade de manter a integridade das soluções tem resultado em novas técnicas para solução de problemas de planejamento florestal. Uma das formas de modelar esse tipo de problema é a utilização de modelos de Programação Inteira (PI). Porém, a formulação de modelos de PI para a resolução de problemas de planejamento florestal pode resultar em modelos de natureza combinatória de difícil solução. Alguns algoritmos como *branch and bound* são utilizados para resolver o problema. No entanto, estes funcionam apenas para instâncias de pequeno porte.

2 O PROBLEMA E SUA IMPORTÂNCIA

As florestas plantadas, bem como as florestas nativas, são importantes para a sociedade por serem responsáveis pela eliminação de grande parte da concentração de óxido de carbono (CO_2) na atmosfera. Além disso, estas florestas também são utilizadas como corredores naturais, dos quais se tornam habitat de várias espécies de animais, provendo abrigo e alimentação. Outras funções importantes das florestas estão relacionadas à água, pois auxilia na manutenção das nascentes e na proteção dos leitos dos rios.

As florestas plantadas fornecem outros benefícios para o ser humano, principalmente por terem se tornado uma das principais fontes de matéria-prima e um importante fator de competitividade para diversos segmentos industriais como: celulose e papel, siderurgia, painéis, móveis, etc., se tornando cada vez mais importantes para o setor econômico do país.

Dados da Indústria Brasileira de Árvores (IBA, 2014) referentes ao ano de 2013 mostram uma pequena evolução do setor florestal em relação ao ano de 2012. Em 2013, o total de área plantada no país foi de 7,60 milhões de hectares, enquanto que em 2012 foram de 7,39 milhões hectares. Em 2013, o consumo brasileiro de madeira de árvores plantadas para uso industrial foi de 185,3 milhões de metros cúbicos (m^3), o que representou aumento de 1,8% em relação ao consumo de 2012. Logo, com o crescimento da produção, se torna essencial um melhor planejamento para se gerar uma maior lucratividade, pensando nisso que se iniciou a pesquisa em torno do Problema de Planejamento Florestal (PPF).

O PPF consiste na escolha da melhor Alternativa de Manejo (AM) para determinadas áreas homogêneas conhecidas como Unidades de Produção (UP's), também chamadas de talhão por profissionais da área de Engenharia Florestal.

Para Pereira (2004), alternativa de manejo (AM) é uma sequência de ações que são realizadas em um dado período de tempo. Neste trabalho, foram consideradas 2 formas de manejo, a condução por reforma, na qual a floresta é totalmente arrancada e novas árvores são plantadas, e o regime de condução por brotação, no qual corta-se a floresta e espera-se que um broto da própria árvore tome seu lugar. O manejo por brotação gera menos trabalho e menores custos, porém, a reforma gera uma produtividade maior (RODRIGUES et al., 1999). Mas, se com o passar do tempo uma unidade de produção vem sofrendo somente manejo por brotação, acaba produzindo menos do que deveria, obrigando assim um manejo por reforma. Por outro lado, as empresas não podem ultrapassar um número máximo de reformas, pois este tipo de manejo gera custos maiores, o que acabaria diminuindo os lucros da empresa.

Já uma UP é uma divisão feita na floresta com a finalidade de melhor identificar as características deste local. As árvores de uma mesma UP possuem atributos

semelhantes, tais como, material genético, idade e classe de produtividade. Outras características comuns de cada UP estão relacionadas com o solo, tais como declive e altitude.

Weintraub et al. (2000), Teixeira (2002) e Goycoolea et al. (2009) destacam as características inerentes aos problemas de planejamento do manejo florestal: natureza combinatória, problema típico de programação inteira, objetivos competitivos e conflitantes e que contêm muitas variáveis aleatórias.

Para melhor ilustrar o problema, a Figura 1 apresenta uma floresta e esta, por sua vez, está dividida em 19 UP's. Como dito anteriormente, cada UP contém características próprias, inclusive o conjunto de AM's. A quantidade de AM's para cada UP depende da idade das árvores, da definição das opções de manejo e do horizonte de planejamento do problema, podendo ser considerada uma sequência de decisões que ocorrem ao longo de um horizonte de planejamento. Tais decisões podem representar, por exemplo, épocas de cortes, plantios, intensidades de desbastes, tecnologias e materiais genéticos a serem utilizados.

Somente uma Alternativa de Manejo (AM) deve ser escolhida para cada uma das UP's tratadas no problema.

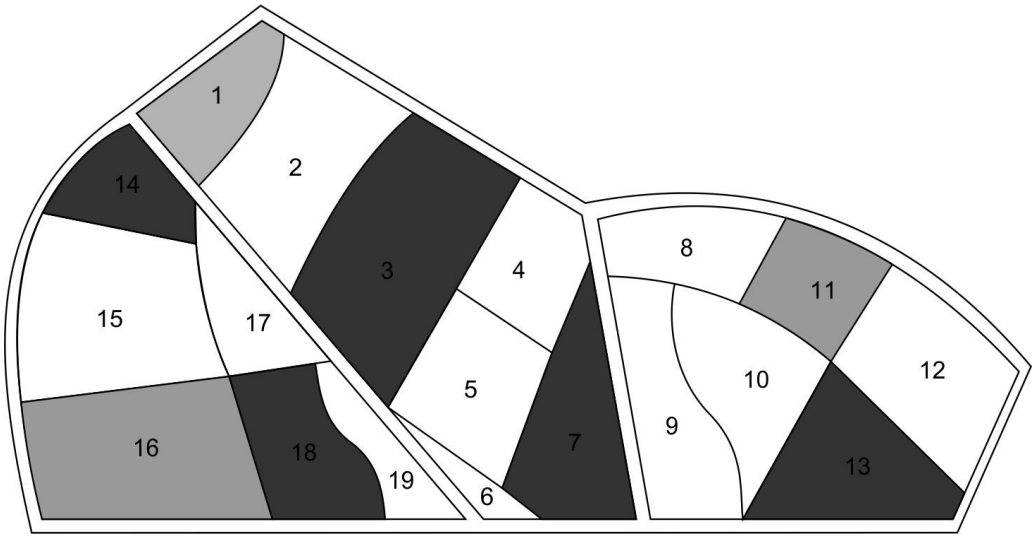


Figura - Área florestal dividida em 19 UP's.
Fonte: Melo et al. (2014).

Para a contínua evolução do setor de florestas plantadas, é necessário que os produtores façam um melhor planejamento da produção considerando as diversas

restrições do problema. Esse tipo de planejamento envolve, normalmente, um grande número de aspectos como: limitações de área, capital e mão de obra; além de atender várias outras restrições como: demanda anual de mercado, regulação da floresta, capacidade de reforma, entre outros. Muitos desses fatores são agravados pelas características próprias do setor florestal, que trabalha com ciclos longos e, além de tudo, a empresa tem que atender exigências legais, ambientais e sociais, o que torna a análise muito complexa (BARROS JUNIOR et al., 2012).

Cabe ao gerente florestal o desafio de decidir qual é o momento ideal de realizar cada intervenção nas unidades de produção de modo a manter a produção constante e ao mesmo tempo obter o máximo retorno financeiro do empreendimento como um todo.

3 OBJETIVOS

A seguir, são descritos o objetivo geral do trabalho e os objetivos específicos.

4 **Objetivo geral**

O objetivo deste trabalho é testar diferentes meta-heurísticas como geradoras de soluções para a CS e utilizá-las como alternativa para resolução de um Problema de Planejamento Florestal. O desempenho dessas meta-heurísticas é comparado com resultados obtidos por outros métodos encontrados na literatura.

5 **Objetivos específicos**

a) Implementar a meta-heurística CS com diferentes meta-heurísticas geradoras de soluções.

b) Aplicar as variações da meta-heurística CS para resolver instâncias do PPF.

c) Avaliar o desempenho dos métodos por meio de experimentos computacionais comparados a outros trabalhos e/ou métodos existentes na literatura.

d) Verificar o funcionamento da técnica de intensificação *Path Relinking* (PR) na variação da CS que obtiver os melhores resultados.

6 REVISÃO DA LITERATURA

No Brasil, vários trabalhos vêm utilizando métodos de otimização para o planejamento dos recursos florestais. A seguir, são apresentados alguns destes trabalhos encontrados na literatura.

Rodrigues (2001) apresenta a Busca Tabu, o *Simulated Annealing*, o Algoritmo Genético e métodos híbridos para a resolução do PPF os resultados obtidos são comparados com os obtidos pelo método de *branch and bound*. Os experimentos realizados comprovaram a utilidade de usar procedimentos heurísticos para resolver o problema. Já em Rodrigues et al. (2003), o PPF é abordado considerando uma restrição inteira, na qual decide-se abater ou não uma UP. Nesse trabalho, foi utilizada a meta-heurística Busca Tabu. Já Rodrigues et al. (2004) focou a calibração do *Simulated Annealing* (SA) para resolver o PPF com o objetivo de superar os resultados anteriores.

Rodrigues et al. (2006) comparou em sua pesquisa dois modelos para o PPF, o Modelo I, e o Modelo II (JOHNSON; SCHEURMAN, 1977). Os autores observaram que o Modelo II possui um número inferior de restrições quando comparado ao Modelo I, porém, o Modelo II se torna mais complexo por possuir restrições que devem ser definidas de forma não-linear.

Gomide (2009) utilizou as meta-heurísticas SA e Algoritmo Genético (AG) na resolução do PPF utilizando restrições de adjacência. Nesses experimentos, provou-se que algoritmos exatos que trabalham com programação linear inteira (PLI) não são capazes de resolver instâncias de grande porte e, portanto, a única opção para resolver essas instâncias é a utilização de métodos heurísticos.

Silva et al. (2009) utilizou a meta-heurística AG para resolver o PPF. No estudo, comprovou-se que a população inicial é o que mais influência na qualidade das soluções, tanto na questão de tempo quanto na aproximação da solução ótima.

Barros Junior et al. (2009) aplicou o método *Greedy Randomized Adaptive Search Procedure* (GRASP) com a técnica de intensificação *Path Relinking* (PR) e obteve resultados satisfatórios, constatando que a técnica PR consegue efetivamente intensificar a qualidade dos resultados.

No trabalho de Barros Junior e Pereira (2011), foi proposto a utilização do AG juntamente com a técnica de intensificação PR. Os bons resultados obtidos nesse trabalho os tornaram base de comparação para outros métodos.

Recentemente, Barros Junior et al. (2012) propôs o uso do *Simulated Annealing* na resolução do problema citado, porém, os resultados obtidos foram inferiores aos já conhecidos.

Na busca por novas opções para resolver o PPF, decidiu-se optar pelo uso da *Clustering Search* (CS), pois esta é uma meta-heurística que está sendo amplamente utilizada para resolução de diversos problemas combinatórios e vem conseguindo alcançar bons resultados. Um dos exemplos de utilização da CS foi na pesquisa de Chaves et al. (2008) na qual a CS foi utilizada juntamente com a SA para resolver o problema de balanceamento e designação de trabalhadores em linha de produção.

Já Almeida e Senne (2010) utilizaram a CS com SA para resolver um problema de localização de hubs com restrições de capacidade. No trabalho de Chaves et al. (2012) foi utilizada a CS com ILS na resolução do problema de minimização de trocas de ferramentas.

Araújo e Mauri (2013) propuseram a utilização da *Simulated Annealing* (SA) com a CS para resolução de um problema de rotação de culturas com restrições de adjacências. E por fim, Rabello et al. (2013) também utilizaram a SA com a CS, porém, nesse caso, para resolver o problema de rotulação cartográfica de pontos.

7 METODOLOGIA

8 MODELO MATEMÁTICO

Para a resolução do PPF, foi utilizado o modelo de Programação Inteira proposto por Johnson e Scheurman (1977), o qual apresenta a seguinte especificação:

Maximizar

$$Z = \sum_{i=1}^{M} \sum_{j=1}^{N_i} VPL_{ij} x_{ij} \tag{1}$$

Sujeito a:

$$\sum_{j=1}^{N_i} x_{ij} = 1 \qquad i = 1,\ldots,M \tag{2}$$

$$V_k \leq \sum_{i=1}^{M} \sum_{j=1}^{N_i} V_{ijk} x_{ij} \qquad k = 0,1,\ldots,H-2 \tag{3}$$

$$CR_k \geq \sum_{i=1}^{M} \sum_{j=1}^{N_i} A_{ijk} x_{ij} \qquad k = 0,1,\ldots,H-2 \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad i = 1,\ldots,M;\ j = 1,\ldots,N_i \tag{5}$$

Em que:

Z = lucro total (soma dos lucros de cada unidade de produção);

M = número total de unidades de produção;

N_i = número total de alternativas de manejo da *i*-ésima unidade de produção;

x_{ij} = representa a decisão de cortar ou não a unidade de produção *i* sob a alternativa de manejo *j*.

V_k = volume total (m³) da demanda a ser atendida no período *k-ésimo* período de tempo;

V_{ijk} = volume total produzido pela unidade de produção *i* sob a alternativa de manejo *j*, no *k*-ésimo período do horizonte de planejamento conforme a idade de corte

naquele período;

H = horizonte de planejamento;

CR_k = capacidade de reforma da floresta no período k;

A_{ijk} = área da i-ésima unidade de produção sob a j-ésima alternativa de manejo, no início do período k e que esteja sob o regime de manejo de reforma;

VPL_{ij} = valor presente líquido (VPL) da unidade de produção i sob a alternativa de manejo j. Esse valor é dado pela diferença dos valores de receitas e custos, descontados a uma taxa determinada, conforme Equação (6):

$$VPL_{ij} = \sum_{k=0}^{n} R_k (1+t)^{-k} - \sum_{k=0}^{n} C_k (1+t)^{-k} \tag{6}$$

R_k = receita no final do período de tempo k;

C_k = custo no final do período de tempo k;

t = taxa de desconto;

n = duração do projeto em número de períodos de tempo.

A função objetivo (1) representa a maximização do lucro, ou seja, a soma dos Valores Presentes Líquidos (VPL) das UP's sobre uma alternativa de manejo escolhida. A restrição (2) garante a escolha de uma única alternativa de manejo j para a unidade de produção i. A restrição (3) representa a imposição de cotas anuais de produção, ou seja, a produção periódica de madeira deve atender a uma demanda preestabelecida. A restrição (4) representa a capacidade máxima de reforma que a empresa florestal é capaz de realizar a cada período. Por fim, a restrição (5) garante que as variáveis de decisão sejam binárias, ou seja, indicam se a unidade de produção i será cortada sob a alternativa de manejo j (x_{ij} = 1), ou não (x_{ij} = 0).

9 REPRESENTAÇÃO DA SOLUÇÃO.

No caso do PPF, pode-se representar uma solução S na forma um vetor, no qual

cada posição representa uma UP diferente. Assim, o vetor terá o número de posições igual ao número de UP's da instância do problema, de acordo com a representação apresentada na Figura 2. Para cada UP no vetor, existe um conjunto de possíveis alternativas de manejo (AM) associadas, mas, para cada UP, apenas uma alternativa *j* deve ser escolhida. No exemplo ilustrado pela Figura 2, para a solução *S*, foram escolhidas as alternativas de manejo 8, 2, 3, ..., 1 e 5 para as UP's 1, 2, 3, ..., 18 e 19 respectivamente.

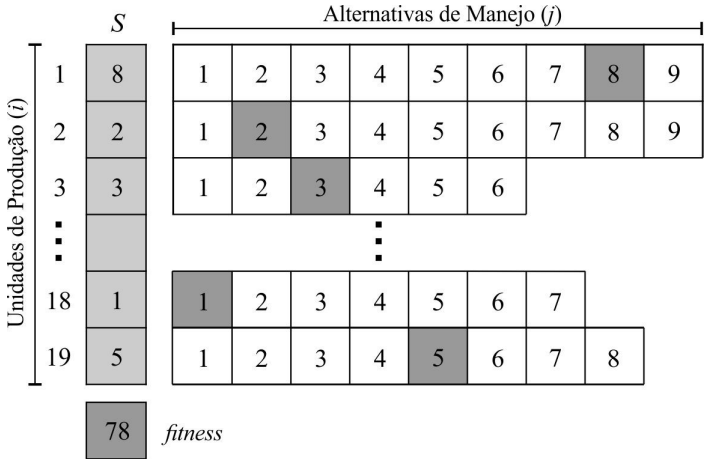
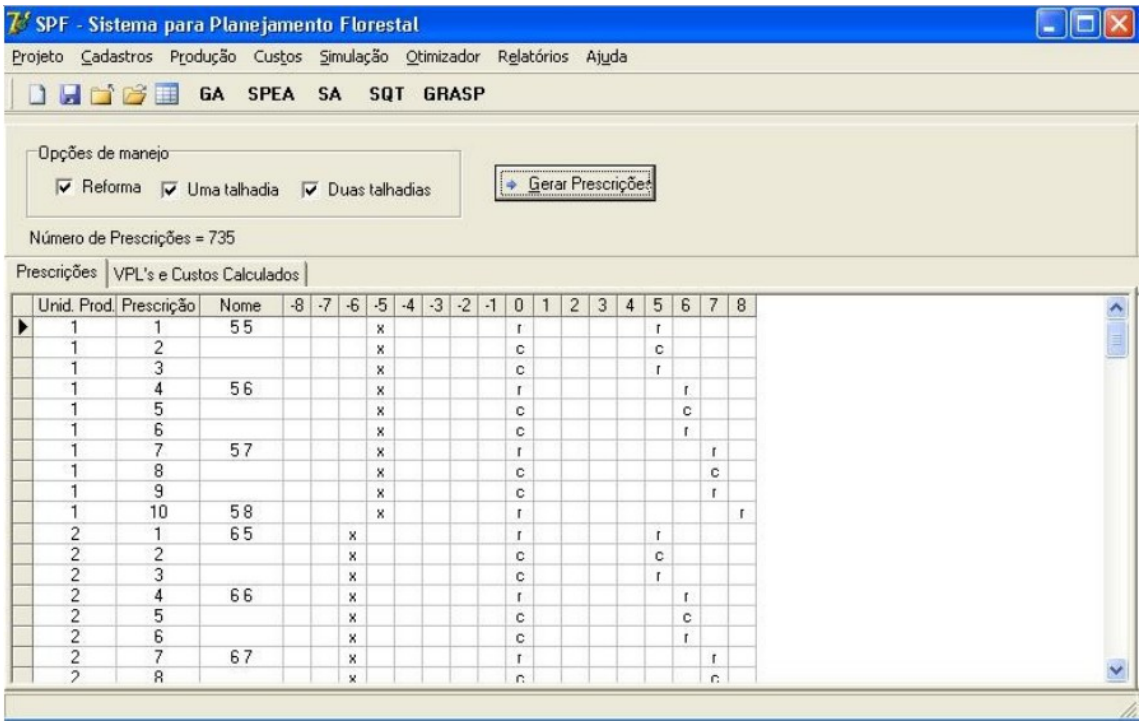


Figura - Representação de uma solução S.
Fonte: Melo et al. (2014).

As AM's consideradas neste trabalho foram obtidas por meio de um protótipo proposto por Teixeira (2002) e adaptado para outros métodos, denominado Gerador de Prescrições (Figura 3). Esse gerador é um *software* que gera um conjunto de AM's para cada UP que compõe uma floresta.



Unid. Prod.	Prescrição	Nome	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
1	1	5 5				x					r					r			
1	2					x					c					c			
1	3					x					c					r			
1	4	5 6				x					r						r		
1	5					x					c						c		
1	6					x					c						r		
1	7	5 7				x					r							r	
1	8					x					c							c	
1	9					x					c							r	
1	10	5 8				x					r								r
2	1	6 5			x						r					r			
2	2				x						c					c			
2	3				x						c					r			
2	4	6 6			x						r						r		
2	5				x						c						c		
2	6				x						c						r		
2	7	6 7			x						r							r	
2	8				x						c							c	

Figura - Gerador de prescrições.
Fonte: Barros Junior et al. (2009).

Uma prescrição é cada alternativa de manejo (AM) em uma unidade de produção (UP). Para cada UP, podem existir centenas de AM's diferentes, que são obtidas pela combinação das idades de corte, opções de reforma da floresta, desbastes ou condução de brotação e quaisquer outras características que se queira. No exemplo da Figura 3, foram geradas 10 prescrições para a UP 1. O objetivo dos métodos propostos é escolher a melhor AM para cada UP, levando em conta todas as restrições impostas pelo problema, e buscando obter o maior Valor Presente Líquido (VPL). A alternativa selecionada na Figura 3 representa que a cada 5 anos a UP sofrerá intervenção, sendo uma reforma no ano "0", que representa o ano atual, e outra reforma no ano 5. O ano no qual a floresta foi plantada é representado pela letra "x".

10 *CLUSTERING SEARCH* (CS).

A CS parte do pressuposto que um processo de busca pode ser otimizado por meio do agrupamento de soluções em *clusters* diferentes e realizando uma busca local somente em *clusters* promissores. A Figura 4 representa o fluxograma da CS.

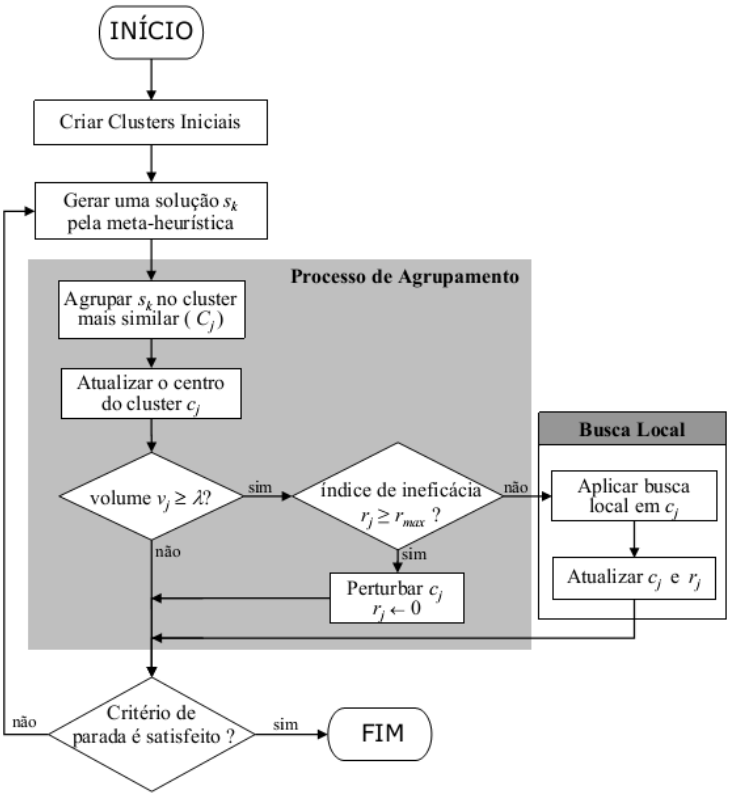


Figura - Fluxograma da CS.
Fonte: Chaves e Lorena (2010).

A CS é um método iterativo que procura dividir o espaço de busca e localizar regiões promissoras por meio do enquadramento das soluções em *clusters*. Um *cluster j* pode ser representado como $C_j = \{c_j, v_j, r_j\}$, sendo c_j, v_j e r_j representações do centro, volume e ineficácia, respectivamente (CHAVES; LORENA, 2010).

O centro do *cluster* (c_j) é uma solução que representará o *cluster* C_j no espaço de soluções. O volume (v_j) do *cluster* C_j, indica o número de soluções que estão alocadas no *cluster* e, quando v_j alcançar um limitante (λ) deve-se utilizar uma busca local neste *cluster*, pois este representa uma região promissora no espaço de soluções. Outro componente importante é o índice de ineficácia (r_j), que indica se o procedimento de busca local está melhorando a solução do centro *cluster* (c_j). Caso r_j alcance o limitante r_{max}, a busca local não está melhorando a solução, portanto é necessário aplicar uma perturbação no centro do *cluster.* O pseudocódigo da CS é apresentado na Figura 5.

Procedimento CS

```
1   Dada (uma solução S);
2   Encontrar (o cluster C_j mais similar a S);
3   v_j ← v_j + 1;
4   Atualizar (o centro do cluster C_j);
5   Se v_j ≥ λ então
6       v_j ← 1;
7       Se r_j ≥ r_max então
8           Aplicar (perturbação em c_j)
9           r_j ← 0;
10      Senão
11          Aplicar (busca local – gerar c_j');
12          Se f(c_j') < f(c_j) então
13              r_j ← r_j + 1;
14          Senão
15              c_j ← c_j';
16              r_j ← 0;
17          Fim-se;
18      Fim-se;
19  Senão
20      V_j ← v_j + 1;
21  Fim-se;
22  Retorna melhor solução encontrada;
```

Figura - Pseudocódigo da CS.
Fonte: Adaptado de Melo et al. (2013).

Em linhas gerais, a CS pode ser descrita basicamente por três partes: (1) uma meta-heurística geradora de soluções; (2) um algoritmo de agrupamento de soluções; e (3) uma heurística de busca local.

Neste trabalho, a criação dos *clusters* iniciais foi realizada por meio das primeiras iterações da meta-heurística geradora. Um número máximo de *clusters* γ é definido e, assim, as primeiras γ soluções geradas são atribuídas como centro dos *clusters*. A partir de então, as três partes básicas da CS são utilizadas, conforme descrito a seguir.

10.1.1 Algoritmo de agrupamento

As soluções geradas pela meta-heurística geradora são posteriormente enviadas para a CS, na qual são agrupadas nos *clusters* mais similares. Para descobrir qual o *cluster* mais similar a uma solução *S*, foi utilizada a distância de *Hamming* (HAMMING, 1950). Assim, como cada solução é representada por meio de um vetor (Figura 2), a distância de *Hamming* é obtida por meio do cálculo das diferentes

alternativas de manejo atribuídas a cada unidade de produção. Logo, quanto mais diferenças existirem entre duas soluções, maior será a “distância” entre elas. Assim, a solução *S* será agrupada ao *cluster* com a menor distância em relação ao seu centro (c_j).

10.1.2 Busca local

O procedimento de busca local utilizado neste trabalho consiste em rotinas de trocas das alternativas de manejo de uma solução. Assim, para cada UP da solução *S*, são testadas todas as AM’s. Após cada troca, uma nova solução é gerada e sua viabilidade é verificada. Caso a solução gerada seja viável e melhor que a solução *S* (parâmetro do procedimento), ela é armazenada para ser comparada com a próxima solução. Ao final do procedimento, a melhor solução gerada é retornada pelo procedimento.

10.1.3 Perturbação

A perturbação é realizada em uma parte aleatória de uma solução, sendo essa parte escolhida de acordo com um índice *d* que é definido de acordo com o número de UP’s. Neste trabalho, foi definido o índice de 20% do total de UP’s. Esse índice foi definido empiricamente. Logo após selecionar essas UP's da solução, as AM's escolhidas para elas são substituídas aleatoriamente. O objetivo dessa perturbação é fazer com que os centros dos *clusters* consigam migrar de região dentro do espaço de soluções, fazendo com que o algoritmo escape de ótimos locais.

10.1.4 Meta-heurísticas geradoras

A meta-heurística geradora é responsável por preparar uma solução e enviá-la para a CS, fazendo com que a solução chegue à CS já lapidada e validada. Nesta pesquisa, foram utilizadas 4 meta-heurísticas geradoras diferentes, as quais são detalhadas a seguir.

10.1.4.1 *Greedy Randomized Adaptive Search Procedure* (GRASP)

A *Greedy Randomized Adaptive Search Procedure* (GRASP) foi proposta por Feo e Resende (1995) e consiste em um procedimento iterativo de duas fases na qual a primeira consiste em construir uma solução inicial e na segunda é realizada uma

busca local nesta solução com objetivo de encontrar melhores soluções.

De forma geral, a meta-heurística GRASP procura combinar da melhor forma possível as características dos algoritmos gulosos e dos procedimentos aleatórios para a construção das soluções iniciais e posteriormente refiná-las por meio de uma fase de busca local (RESENDE; RIBEIRO, 2002).

A primeira fase da GRASP consiste em construir uma solução inicial. Primeiramente, criou-se um conjunto *CA* com *N* soluções aleatórias para o problema e, em seguida, essas soluções foram ordenadas de acordo com o valor de sua função objetivo. A solução inicial *S* foi definida a partir da escolha aleatória de uma AM para cada UP nas *n* (parâmetro calibrado) primeiras soluções do conjunto *CA* (BARROS JUNIOR et al., 2009).

Logo após a primeira fase é realizada uma busca local na solução *S* na qual é gerada uma nova solução *S*'. Esta busca local é a mesma utilizada pela CS, descrita na Seção 3.3.2. Decidiu-se utilizar a mesma busca local no GRASP e na CS porque, durante os testes realizados, esta obteve os melhores resultados. Após realizada a busca local, a solução *S*' é comparada com a melhor solução encontrada até o momento *S**. Caso *S*' seja melhor que *S**, *S*' se tornará a melhor solução. No fim do procedimento, a melhor solução (*S**) é retornada.

A Figura 6 apresenta um pseudocódigo da meta-heurística GRASP implementada.

```
Procedimento GRASP (n) // n = numero de AM's
 1    f ← 0;
 2    Enquanto (critériodeparada) faça
 3        S ← construçãoParcialmenteGulosa(n);
 4        S' ← BuscaLocal (S);
 5        Se ( f(S')>f (S*)) então
 6            S* ← S;
 7            f* ← f(S')
 8        Fim-se;
 9    Fim-enquanto;
10    S ← S* ;
11    Retorne S;
Fim GRASP;
```

Figura - Pseudocódigo da GRASP.
Fonte: Feo e Resende (1995).

10.1.4.2 *Simulated Annealing* (SA)

O algoritmo SA, introduzido por Kirkpatrick et al. (1983), foi criado com base no processo de recozimento de materiais sólidos. Na analogia, o metal inicia com uma alta temperatura, a qual vai diminuindo gradativamente, até atingir o ponto de solidificação, que é caracterizado como o estado de energia mínima.

A SA procura explorar novas áreas no espaço de soluções do problema em um curto período de tempo e com pouco esforço. O algoritmo implementado utiliza como entrada uma solução gerada aleatoriamente (*S*). Logo em seguida, é gerado *S'* a partir de uma mutação em *S*, e essa mutação consiste em escolher 30% das UP's de *S* e trocar aleatoriamente as suas AM's. Após gerar *S'*, verifica-se a variação de energia. Neste trabalho, a variação de energia é calculada através da diferença entre as funções objetivos de *S'* e *S* e, caso a variação de energia seja positiva, a solução *S* recebe *S'* e, caso contrário, é escolhido um número aleatório *r* no conjunto [0,1] e o valor de *w* é calculado. Neste caso, *w* representa a probabilidade de aceitar soluções piores que a solução atual.

A fórmula para o cálculo de *w* é dada pelo exponencial do valor da função objetivo da solução *S* menos o valor da função objetivo de S' dividido pela temperatura. Se o resultado de *w* for maior que o *r,* escolhido anteriormente, *S* recebe *S'*, senão S' é eliminada e retorna ao passo 2. No final, o algoritmo retorna *S.* O algoritmo básico da meta-heurística SA é apresentado na Figura 7.

Procedimento SA (*S, S', T*)
1 **Enquanto não** CritérioParada **faça**
2 S' ← S; //gerar S' a partir de S
3 $\Delta E = f(S') - f(S)$ //Variação de energia
4 $\Delta E > 0$ **Se** **então**
5 S ← S';
6 **Senão**
7 Gerar $r \in$ Unif[0,1];
8 $r < w = e^{-(f(S') - f(S)/kT)}$ **Se** **então**
9 S ← S';

```
1                T ← a * T; // a = índice de redução
0
11            Fim-se;
1         Fim-senão;
2
1     Fim-Enquanto;
3
Fim SA;
```

Figura - Pseudocódigo do algoritmo SA.
Fonte: Barros Junior et al. (2012).

10.1.4.3 *Iterated Local Search* (ILS)

A *Iterated Local Search* (ILS) é uma meta-heurística que se baseia na ideia de que um procedimento de busca local pode ser melhorado, gerando-se novas soluções de partida, as quais são obtidas por meio de perturbações em uma solução ótima local (LOURENÇO et al., 2002).

Nesse método, o procedimento de busca local tem por finalidade melhorar a solução inicial e também aquelas produzidas após perturbações de soluções ótimas locais (BRITO et al., 2009). Dessa forma, a perturbação deve ser calibrada de forma a permitir que a busca local explore diferentes soluções, além de evitar um reinício aleatório.

A Figura 8 apresenta um pseudocódigo básico do algoritmo da ILS, que possui 4 etapas principais. A primeira etapa é responsável pela criação de uma solução inicial S_1 que servirá de ponto de partida para o método. Na segunda etapa é realizada uma perturbação na solução S_1 obtendo uma nova solução S_2. Essa perturbação é realizada em uma parte aleatória da solução, e assim as AM's de cada UP selecionada são trocadas também de forma aleatória. A parte da solução a ser alterada é escolhida de acordo com um índice *d* (índice de destruição) que equivale a certa porcentagem do número de UP's.

Procedimento *ILS* (*CritérioParada*)

1 $S_1 \leftarrow$ Construção_Solução_Inicial;
2 $S^* \leftarrow S_1$; *//Melhor solução encontrada*
3 **enquanto não** *CritérioParada* **faça**
4 $S_2 \leftarrow$ Perturbação(S_1,*d*);
5 $S_2 \leftarrow$ Busca_Local(S_2);
6 Criterio_Aceitação(S^*,S_1,S_2);
7 **fim-enquanto**;
8 retorne *S*;

fim *ILS*;

Figura - Pseudocódigo da ILS.
Fonte: Lourenço et al. (2002).

Na etapa seguinte, é aplicada uma busca local em S_2 que tem o objetivo de vasculhar o espaço de soluções em busca da melhor solução. A etapa final responde pelo critério de aceitação, sendo que as soluções S_1, S_2 são comparadas com a melhor solução obtida até o momento. Se alguma delas for melhor esta se tornará a melhor solução e, caso contrário, a solução corrente é descartada e a melhor solução continua sendo a antiga. Nesse momento, é testada a condição de parada, se for satisfeita a melhor solução atual é retornada, caso contrário volta-se a segunda etapa. Com exceção da primeira etapa, todas as demais são executadas iterativamente enquanto o critério de parada não é atendido. Após as iterações, o algoritmo retorna a melhor solução obtida durante toda sua a execução.

10.1.4.4 Algoritmo Genético (AG)

O Algoritmo Genético (AG), proposto por Holland (1975), é baseado nos processos de melhoria genética, nos quais visam criar novos indivíduos através de cruzamentos genéticos, visando melhorar a qualidade e a variabilidade entre indivíduos.

O algoritmo genético trabalha com o conceito de população, portanto, primeiramente é gerada uma quantidade *n* de indivíduos (soluções) de forma aleatória. Esses indivíduos são colocados em uma população que se tornará a população inicial (P_1), sendo que o melhor indivíduo gerado para a população inicial se torna a melhor solução (S^*). Após criada a população inicial e gravada a melhor solução, é realizada a etapa de seleção. Essa etapa é responsável por selecionar 2 soluções (*s1 e s2*) na população P_1 de forma aleatória, e esses indivíduos selecionados são adicionados em uma nova população (P_2). Na Figura 9 é apresentado um pseudocódigo do AG implementado.

Procedimento AG (CritérioParada)
1 $P_1 \leftarrow$ Inicializa População Inicial Ramdomicamente
2 $S^* \leftarrow P_1$; //melhor solução encontrada
3 **Enquanto não** CritérioParada **faça**
4 $P_2 \leftarrow$ Seleção (P_1) ;// seleciona 2 indivíduos de P_1
5 $P_2 \leftarrow$ Crossover (P_2);

```
6        P2 ← Mutação (P2);
7        P1 ← Survive (P1 , P2); //seleciona melhores indivíduos
8        S* ← P1 ; //melhor solução encontrada
9     Fim-Enquanto;
1     Retorne S*;
0
Fim AG;
```

Figura - Pseudocódigo do AG.
Fonte: Barros Junior e Pereira (2011).

Os indivíduos selecionados na fase anterior são utilizado na fase de cruzamento (*crossover*). O *crossover* consiste em combinar características dos indivíduos selecionados (*s1* e *s2*). Para combinar os indivíduos, deve-se primeiramente escolher a quantidade de pontos de cortes que serão realizados nas soluções. Nesta pesquisa, foi utilizado apenas um ponto de corte, o qual é realizado em 20% da solução. As UP's selecionadas pelo ponto de corte são trocadas, ou seja, as AM's que foram escolhidas para as UP's da solução *s1* são trocadas pelas AM's das UP's selecionadas da solução *s2* e vice versa. Por fim, os novos indivíduos gerados são inseridos na população P_2.

Logo após o *crossover*, é realizada uma mutação nos novos indivíduos gerados. A mutação consiste em escolher 2 UP's na solução e substituir aleatoriamente as AM's destas UP's. Os indivíduos resultantes da fase de mutação são novamente inseridos na população P_2.

O próximo passo do processo é o *Survive* que é a etapa na qual são eliminados os indivíduos que possuem o menor valor para a função objetivo. Neste processo, as soluções que fazem parte das populações P_1 e P_2 serão colocadas em apenas uma população e são ordenadas pelo valor da função objetivo de forma decrescente. Os *n* primeiros elementos são selecionados para formar a nova população inicial (P_1), e os demais são eliminados. No passo 8 é verificado se a melhor solução da população P_1 é melhor que S^*. Em caso positivo, S^* recebe a melhor solução de P_1. No final, o algoritmo retorna S^*, que é a melhor solução encontrada.

11 *PATH RELINKING* (PR)

O algoritmo *Path Relinking* (PR), proposto por Glover (1999), é uma técnica de intensificação, na qual procura melhores soluções explorando trajetórias existentes entre duas soluções.

O PR recebe como entrada duas soluções, uma solução de origem (S_o) e uma solução guia (S_g). O objetivo é transformar a solução S_o em S_g e gerar, nesse caminho, novas soluções. Ao final o método deve retornar a melhor solução dentre todas as soluções geradas.

A cada iteração do algoritmo ele troca a alternativa de manejo de uma UP da S_o pela alternativa de manejo da mesma UP de S_g (se elas forem diferentes) e a cada iteração a melhor solução é guardada. As iterações do algoritmo terminam quando a solução S_o for igual a S_g.

12 RESULTADOS

Os parâmetros das meta-heurísticas geradoras foram escolhidos de acordo com os parâmetros encontrados na literatura, desta forma os parâmetros da GRASP seguem o trabalho de Barros Junior et al. (2009). Já o *Simulated Annealing* (SA) segue o trabalho de Barros Junior et al (2012) e o Algoritmo Genético (AG) segue o trabalho de Barros Junior e Pereira (2009). Por fim, o *Iterated Local Search* (ILS) foi baseado em Melo et al. (2013). Os valores dos parâmetros utilizados para cada um desses métodos são apresentados nas Tabelas 1 a 4.

Tabela - Parâmetros da CS-GRASP.

CS-GRASP		
GRASP	Taxa para lista restrita de candidatos	15%
	Lista restrita de candidatos	100%
	Soluções Escolhidas	100
CS	*Clusters*	7
	Volume	5
	Ineficácia	3

Tabela - Parâmetros da CS-SA

CS-SA		
SA	Temperatura Inicial	1000000
	Taxa de resfriamento	0,3%
	Congelamento	50
	Taxa alteração	30%
CS	*Clusters*	5
	Volume	5
	Ineficácia	3

Tabela - Parâmetros da CS-AG

CS-AG		
AG	Cruzamento (1 ponto)	80%
	Taxa de Sobreviventes	5%
	Mutação	10%
	Gene	10%
CS	*Clusters*	5
	Volume	5
	Ineficácia	5

Tabela - Parâmetros da CS-ILS

CS-ILS		
ILS	Taxa para lista restrita de candidatos	15%
	Lista restrita de candidatos	100%
	Soluções Escolhidas	100
	Taxa de destruição	20%
CS	*Clusters*	5
	Volume	10
	Ineficácia	5

Já para a CS, os parâmetros foram definidos empiricamente, por meio de diversos experimentos computacionais. O critério de parada para ambos os métodos foi baseado no tempo de execução de acordo com a Tabela 5, assim como nos demais trabalhos utilizados como comparação.

Tabela - Descrição das instâncias utilizadas.

Instância	Classe	Nº UP's	Horizonte	Nº AM's	Tempo
1	A	50	9 anos	735	5 min.
2	A	50	9 anos	735	5 min.
3	A	50	9 anos	735	5 min.
4	A	50	14 anos	2466	10 min.
5	A	50	14 anos	2466	15 min.
6	A	50	14 anos	2466	10 min.
7	B	100	9 anos	1533	20 min.
8	B	100	9 anos	1533	20 min.
9	B	100	14 anos	5094	20 min.
10	B	100	14 anos	5094	30 min.

Para a realização dos experimentos computacionais, foram utilizadas 10 instâncias (Tabela 1) baseadas em dados reais e em dados obtidos na literatura (TEIXEIRA, 2002). Mais detalhes sobre as classes, UP's, horizonte e número de AM's descritas na Tabela 2 são apresentados em Barros Junior et al. (2009).

O tempo para resolução de cada instância é definido pela fórmula $(M \times \sqrt{m})/4$, na qual, M representa o número total de UP's e m representa o número total de AM's disponíveis para a instância. O resultado é dado em segundos e posteriormente

convertido para minutos e está apresentado na última coluna da Tabela 5. Essa fórmula segue o modelo encontrado em Barros Junior et al. (2012).

A principal diferença entre as instâncias que contêm o mesmo número de AM's está relacionada com os valores utilizados na definição dos níveis de demanda anual (Tabela 6). Esses valores foram considerados na restrição (2) do problema.

Tabela - Volume total (m³) da demanda a ser atendida no período *k*.

Período (k)	Instâncias									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	500.000	550.000	575.000	575.000	575.000	575.000	1.000.000	1.300.000	900.000	1.100.000
1	500.000	550.000	575.000	575.000	575.000	575.000	1.000.000	1.300.000	900.000	1.100.000
2	600.000	650.000	675.000	575.000	675.000	675.000	1.200.000	1.350.000	1.000.000	1.200.000
3	700.000	750.000	775.000	575.000	675.000	675.000	1.300.000	1.350.000	1.100.000	1.300.000
4	800.000	850.000	875.000	575.000	675.000	675.000	1.400.000	1.400.000	1.200.000	1.400.000
5	800.000	850.000	875.000	575.000	675.000	675.000	1.400.000	1.450.000	1.200.000	1.400.000
6	800.000	850.000	875.000	575.000	675.000	700.000	1.500.000	1.500.000	1.300.000	1.500.000
7	900.000	950.000	975.000	575.000	675.000	700.000	1.500.000	1.500.000	1.300.000	1.500.000
8	900.000	950.000	975.000	575.000	675.000	700.000	1.550.000	1.550.000	1.350.000	1.550.000
9	-	-	-	575.000	675.000	700.000	-	-	1.350.000	1.550.000
10	-	-	-	575.000	675.000	700.000	-	-	1.400.000	1.600.000
11	-	-	-	575.000	675.000	700.000	-	-	1.400.000	1.600.000
12	-	-	-	575.000	675.000	700.000	-	-	1.500.000	1.700.000
13	-	-	-	575.000	675.000	700.000	-	-	1.500.000	1.700.000

Outra diferença entre as instâncias que contêm o mesmo número de AM's está relacionada à capacidade de reforma (Tabela 7), considerada na restrição (4) do problema.

Tabela - Capacidade de reforma (ha) a ser atendida no período *k*.

Período (k)	Instâncias									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	6.000	7.000	8.000	8.000	8.000	8.000	7.500	8.500	7.500	7.500

1	6.500	7.500	8.500	8.000	8.000	8.000	7.500	8.500	7.500	7.500
2	7.000	8.000	9.000	9.000	9.000	8.000	7.500	8.500	7.500	7.500
3	7.500	8.500	9.500	10.000	10.000	8.000	7.500	8.500	7.500	7.500
4	8.000	9.000	10.000	11.000	11.000	8.000	7.500	8.500	7.500	7.500
5	8.500	9.500	10.500	11.000	11.000	8.000	7.500	8.500	7.500	7.500
6	9.000	10.000	11.000	11.000	11.000	8.000	7.500	8.500	7.500	7.500
7	9.500	10.500	11.500	12.000	12.000	8.000	7.500	8.500	7.500	7.500
8	10.000	11.000	12.000	12.000	12.000	8.000	7.500	8.500	7.500	7.500
9	-	-	-	12.000	12.000	8.000	-	-	7.500	7.500
10	-	-	-	12.000	12.000	8.000	-	-	7.500	7.500
11	-	-	-	12.000	12.000	8.000	-	-	7.500	7.500
12	-	-	-	12.000	12.000	8.000	-	-	7.500	7.500
13	-	-	-	12.000	12.000	8.000	-	-	7.500	7.500

Os métodos propostos nesta pesquisa foram executados 10 vezes para cada instância, obedecendo ao tempo estipulado acima. É importante ressaltar que os algoritmos foram codificados na linguagem *Object* Pascal e os testes foram efetuados em um microcomputador Intel Core 2 Duo 2.20GHz com 4GB de memória RAM.

A Tabela 8 apresenta os resultados obtidos pelos métodos CS-GRASP (*Clustering Search* com GRASP), CS-SA (*Clustering Search* com *Simulated Annealing*), CS-ILS (*Clustering Search* com ILS) e CS-AG (*Clustering Search* com Algorítimo Genético), propostos nesta pesquisa.

Observa-se que o CS-SA obteve os melhores resultados em 6 das 10 instâncias testadas. O CS-AG obteve melhores resultados em outras 3 e o CS-GRASP foi melhor em apenas uma das instâncias. Já a CS-ILS não conseguiu melhor resultado em nenhuma das instâncias, mas obteve média melhor que o CS-GRASP.

Tabela - Comparação das soluções obtidas pelas meta-heurísticas propostas.

Instância	CS-GRASP	CS-SA	CS-ILS	CS-AG
1	180.849.191,58	181.154.935,93	180.544.282,35	**181.579.454,85**
2	178.445.048,02	178.108.957,82	177.870.887,46	**178.568.704,79**
3	**176.764.552,98**	174.721.515,29	175.589.876,35	173.649.145,79
4	557.337.475,57	**565.692.445,82**	560.085.359,77	562.269.120,61
5	550.948.367,05	**554.069.694,46**	552.385.227,43	551.306.263,46
6	258.296.804,37	261.157.378,72	259.000.615,57	**261.396.492,15**

7	254.814.613,12	**257.123.247,63**	256.585.336,21	256.139.257,23
8	254.704.255,17	**256.922.817,88**	256.144.408,10	256.039.037,71
9	395.186.928,49	**397.567.269,27**	395.307.241,99	397.296.274,15
10	392.928.546,40	**396.290.224,41**	394.243.080,80	394.823.654,50
Média	320.027.578,27	**322.280.848,72**	320.775.631,60	321.306.740,52

De acordo com esses resultados, o *Simulated Annealing* (SA) se mostrou como a melhor escolha quando se trata de uma meta-heurística geradora de soluções para a *Clustering Search* (CS).

Na Tabela 9 são apresentados os valores de desvio padrão obtidos pelos métodos propostos. Ao observar essa tabela, pode-se perceber que a meta-heurística CS-GRASP obteve um melhor desvio padrão em 7 das 10 instâncias testadas, o que mostra que o método gera sempre soluções muito próximas uma da outra. Já a CS-SA, que obteve a melhor média geral, obteve o melhor desvio padrão em apenas 2 das 10 instâncias. A CS-AG foi melhor em apenas uma instância enquanto a CS-ILS não conseguiu ser melhor em nenhuma das instâncias. Esses resultados indicam que as meta-heurísticas propostas são confiáveis, porém a CS-GRASP consegue gerar sempre soluções mais próximas e com valores aceitáveis.

Tabela - Comparação do desvio padrão obtido pelos métodos propostos.

Instância	CS-GRASP	CS-SA	CS-ILS	CS-AG
1	951.789,27	1.338.852,98	1.173.391,54	**683.048,71**
2	1.433.223,74	**1.179.009,98**	2.476.727,93	1.265.228,96
3	**1.712.399,75**	2.902.034,54	1.896.868,77	3.415.228,36
4	**1.339.323,47**	2.161.194,23	1.571.261,87	2.046.934,70
5	**2.048.027,36**	3.383.591,52	2.939.029,59	3.548.303,72
6	**1.007.315,42**	1.611.815,07	1.016.226,30	1.283.721,01
7	1.299.029,03	**1.099.989,02**	1.754.278,40	1.706.252,62
8	**1.410.949,05**	1.986.716,17	3.195.346,70	2.234.091,77
9	**541.477,75**	550.708,43	1.677.307,83	637.723,54
10	**661.029,39**	730.746,73	695.354,99	1.042.775,92
Média	**1.240.456,42**	1.694.465,87	1.839.579,39	1.786.330,93

Na Tabela 10 são apresentados os resultados obtidos pela CS-SA juntamente com os resultados obtidos pela GRASP proposta por Barros Junior et al. (2009) e o AG proposto em Barros Junior e Pereira (2011), além do ILS (*Iterated Local Search*)

visto em Melo et al. (2013).

Tabela - Comparação das soluções obtidas com a literatura.

Instância	GRASP	AG	ILS	CS-SA
1	170.568.901,33	180.550.495,45	169.481.636,02	**181.154.935,93**
2	168.005.501,97	**178.330.908,66**	167.639.487,08	178.108.957,82
3	165.736.524,15	**175.373.017,91**	165.349.120,37	174.721.515,29
4	520.454.711,87	561.178.394,53	518.305.492,92	**565.692.445,82**
5	511.506.009,49	552.133.334,04	510.175.078,03	**554.069.694,46**
6	238.477.829,62	260.424.178,69	242.221.520,45	**261.157.378,72**
7	237.317.743,93	256.316.557,36	240.994.388,04	**257.123.247,63**
8	236.504.492,97	255.980.694,39	240.068.244,64	**256.922.817,88**
9	362.530.132,21	394.743.106,75	365.912.324,78	**397.567.269,27**
10	359.728.766,18	393.383.603,29	364.280.139,78	**396.290.224,41**
Média	297.083.061,37	320.841.429,11	298.442.743,21	**322.280.848,72**

A CS-SA proposta neste trabalho obteve melhores resultados em 8 das 10 instâncias testadas, ficando o AG melhor em outras 2 instâncias, porém nestas 2 instâncias a CS-GRASP conseguiu resultados melhores que o AG. A CS-SA obteve uma média final 8,48% maior do que a média obtida pelo GRASP (BARROS JUNIOR et al., 2009), e obteve a média final 0,45% maior que a média alcançada pelo AG (BARROS JUNIOR; PEREIRA, 2011).

Na Tabela 11 é apresentado o desvio padrão obtido por meio de cada método.

Tabela - Comparação do desvio padrão obtido com a literatura.

Instância	GRASP	AG	ILS	CS-GRASP
1	1.458.827,85	2.006.727,10	1.435.756,33	**951.789,27**
2	**1.304.077,70**	1.947.806,58	1.396.218,97	1.433.223,74
3	**1.251.111,04**	3.280.134,56	2.588.642,94	1.712.399,75
4	2.222.536,24	2.595.466,75	7.949.461,71	**1.339.323,47**
5	3.550.079,57	3.628.064,57	5.482.554,66	**2.048.027,36**
6	2.087.428,34	1.577.986,04	2.037.091,42	**1.007.315,42**
7	1.823.102,93	2.870.805,05	5.136.751,37	**1.299.029,03**
8	2.840.508,35	2.886.400,50	**1.408.773,65**	1.410.949,05
9	2.242.312,56	1.586.728,04	4.183.461,67	**541.477,75**
10	1.413.780,56	1.435.932,69	3.057.853,64	**661.029,39**
Média	2.019.376,51	2.381.605,19	3.467.656,64	**1.240.456,42**

Os resultados obtidos pela CS-GRASP superaram as demais meta-heurísticas em 7 das 10 instâncias testadas, sendo que a GRASP foi a melhor em 2 e o ILS melhor em outra.

Portanto, a CS-SA foi a meta-heurística que conseguiu alcançar a melhor média final, enquanto a CS-GRASP conseguiu gerar um melhor desvio padrão, isto considerando as meta-heurísticas testadas até o momento.

Como a meta-heurística CS-SA obteve o melhor desempenho entre todas as meta-heurísticas, decidiu-se intensificar os resultados obtidos pela meta-heurística por meio da técnica de intensificação PR, com o intuito de tentar obter melhores resultados, os quais são exibidos na Tabela 12.

Tabela - Médias obtidas pela CS-SA-PR comparado com a CS-SA.

Instância	CS-SA	CS-SA-PR
1	181.154.935,93	**181.397.954,38**
2	178.108.957,82	**178.721.972,86**
3	174.721.515,29	**176.512.225,00**
4	565.692.445,82	**566.004.265,24**
5	554.069.694,46	**556.094.019,05**
6	261.157.378,72	**261.450.720,91**
7	257.123.247,63	**257.847.050,39**
8	**256.922.817,88**	255.367.349,90
9	397.567.269,27	**397.920.295,99**
10	**396.290.224,41**	396.085.782,44
Média	322.280.848,72	**322.740.163,62**

Por meio da Tabela 8, pode-se observar que a técnica de intensificação PR melhorou o resultado obtido em 8 das 10 instâncias testadas.

A Tabela 13 apresenta o desvio padrão alcançado pela CS-SA-PR (CS-SA com *Path Relinking*) comparado com a CS-SA e a CS-GRASP, que possui os melhores resultados até o momento.

Tabela - Desvio padrão obtido pela CS-SA-PR.

Instânci	CS-GRASP	CS-SA	CS-SA-PR

a			
1	**951.789,27**	1.338.852,98	1.080.110,05
2	1.433.223,74	**1.179.009,98**	1.720.824,00
3	**1.712.399,75**	2.902.034,54	1.761.947,50
4	**1.339.323,47**	2.161.194,23	2.775.527,46
5	**2.048.027,36**	3.383.591,52	3.484.922,34
6	**1.007.315,42**	1.611.815,07	1.229.870,93
7	1.299.029,03	**1.099.989,02**	1.312.584,41
8	**1.410.949,05**	1.986.716,17	1.718.435,85
9	541.477,75	550.708,43	**458.366,21**
10	661.029,39	730.746,73	**552.711,23**
Média	**1.240.456,42**	1.694.465,87	1.609.530,00

A técnica PR obteve uma pequena melhora no desvio padrão quando comparado com o CS-SA, sendo melhor em 6 das 10 instâncias. Porém, no geral, o método CS-GRASP ainda obtém o melhor desvio padrão em 6 das 10 instâncias testadas, o CS-SA foi melhor em 2 e o CS-SA-PR em outras 2 instâncias do problema.

13 CONCLUSÕES

Este trabalho apresentou a aplicação da meta-heurística híbrida *Clustering Search* (CS) utilizando as meta-heurísticas *Iterated Local Search* (ILS), formando a CS-ILS, *Greedy Randomized Adaptive Search Procedure* (GRASP), que forma a CS-GRASP, *Simulated Annealing* (SA), que forma a CS-AS, e Algoritmo Genético (AG), formando a CS-AG, como geradoras de soluções para resolver um Problema de Planejamento Florestal (PPF). Além disso, também foi testada a utilização da técnica de intensificação *Path Relinking* (PR) para tentar melhoras os resultados obtidos.

A CS mostrou-se eficaz na resolução do PPF, conseguindo soluções melhores que as obtidas na literatura. Notou-se também que o SA foi a melhor escolha quando se trata de meta-heurística geradora para a CS, pelo menos para a resolução do PPF em questão.

A técnica de intensificação PR mostrou ser uma forma eficiente de se obter melhores soluções para o PPF, sendo capaz de intensificar de forma considerável as soluções obtidas pelas meta-heurísticas.

Assim, a utilização da CS com ILS, GRASP, SA e AG demostraram serem boas

alternativas para resolução do PPF, principalmente se combinadas com a técnica de intensificação PR.

14 REFERÊNCIAS

ALMEIDA, G. W. E SENNE, E. L. F. Metaheurística hibrida com busca por agrupamento Aplicado ao problema de localização de hubs com Restrições de capacidade, *Anais do XLII SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2010.

ARAÚJO, D. L. E MAURI, G. R. Clustering search para resolução de um problema de rotação de culturas com restrições de adjacências, *Anais do XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional,* 2013.

BARROS JUNIOR, A. A.; ARROYO, J. E. C.; SANTOS, H. N. GRASP com path relinking para um problema de planejamento florestal, *Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2009.

BARROS JUNIOR, A. A.; PEREIRA, G.W. Aplicação da metaheurística algoritmo genético com path relinking para a solução de um problema de planejamento florestal, *Anais do XVIII SIMPEP - Simpósio de Engenharia de Produção,* 2011.

BARROS JUNIOR, A. A.; PEREIRA, G. W.; MAURI, G. R.; MELO, R. S. Aplicação da meta-heurística Simulated annealing para a solução de um problema de planejamento florestal, *Anais do XIX SIMPEP - Simpósio de Engenharia de Produção,* 2012.

BRITO, J. A. M.; MONTENEGRO, F. M. T.; OCHI, L. S. Um algoritmo ILS para melhoria de eficiência da estratificação estatística, *Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional,* 2009.

CHAVES, A. A.; MIRALLES, C.; LORENA, L. A. N. Uma metaheurística híbrica aplicada ao problema de balanceamento e designação de trabalhadores em linha de produção, *Anais do XL SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2008.

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

CHAVES, A. A.; SENNE, E. L. F.; YANASSE, H. H. Uma nova heurística para o problema de minimização de trocas de ferramentas, *Gestão & Produção*, 19, 1,17-30, 2012.

FEO, T. A.; RESENDE, M. G. C. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6, 109-133, 1995.

GLOVER, F. Scatter search and path relinking. *New ideas in optimization*, 297-316, 1999.

GOMIDE, L. R. *Planejamento florestal espacial*, Tese (Doutorado em Ciências Florestais) Universidade Federal do Paraná - UFPR, 2009.

GOYCOOLEA, M.; MURRAY, A.; VIELMA, J. P.; WEINTRAUB, A. Evaluating approaches for solving the area restriction model in harvest scheduling*. Forest Science*, 55, 149-165, 2009.

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

HOLLAND, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*.Oxford, England: U Michigan Press, 1975.

INDÚSTRIA BRASILEIRA DE ÁRVORES (IBA), ano base 2013, (http://www.bracelpa.org.br/shared/iba_2014_pt.pdf), 2014.

JOHNSON, K. N.; SCHEURMAN, H. L. Techniques for prescribling optimal timber harvest and investment under different objectives, discussion and synthesis, *Forest Science Monograph*, USA, 1977.

KIRKPATRICK JR., S.; GELATT, C.; VECCHI, M. Optimization by simulated annealing. *Decision Science* 220(4598), p. 498-516, 1983.

LOURENÇO, H. R.; MARTIN, O.; STUTZLE, T. *Iterated local search*. In: Glover, F.; Kochenberger, G. (Eds.), *Handbook of Metaheuristics*. Kluwer Academic Publishers, Norwell, MA, 321-353, 2002.

MELO R. S.; BARROS JUNIOR, A. A.; MAURI, G. R. Clustering search com iterated local search para resolução do problema de planejamento florestal, *Anais do XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2013.

MELO R. S.; BARROS JUNIOR, A. A.; MAURI, G. R. Resolução de um problema de planejamento florestal via clustering search utilizando a meta-heurística GRASP como geradora de soluções, *Anais do XLVI SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2014.

PEREIRA G.W. Aplicação da Técnica de Recozimento Simulado em Problemas de Planejamento Florestal Multiobjetivo (Mestrado em Ciência da Computação) Universidade Federal de Minas Gerais – UFMG, 2004.

RABELLO, R. L.; MAURI, G. R.; RIBEIRO, G. M.; LORENA, L. A. N. Meta-heurística Clustering search aplicada ao problema de Rotulação cartográfica de pontos, *Anais do XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2013.

RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: Fred Glover e Gary A. Kochenberger (eds.), *Handbook of Metaheuristics*, 219-250. Kluwer Academic Publishers, Norwell, MA, 2002.

RODRIGUES, F. L.; LEITE, H. G.; SILVA, M. L.; GOMES, A. N. Determinação de estratégias ótimas de reforma, condução da brotação e compra de terras, utilizando programação linear. *Revista Árvore*, v. 23, n. 2, p. 169-186, 1999.

RODRIGUES, F. L. *Metaheurística e sistema de suporte à decisão no gerenciamento de recursos florestais*, 255p. Tese (Doutorado em Ciência Florestal), Universidade Federal de Viçosa - UFV, 2001.

RODRIGUES, F. L.; LEITE, H. G.; SANTOS, H. N.; SOUZA, A. L. Solução de problemas de planejamento florestal com restrições de inteireza utilizando busca tabu, *Revista Árvore*, 27, 701-713, 2003.

RODRIGUES, F. L.; LEITE, H. G.; SANTOS, H. N.; SOUZA, A. L.; RIBEIRO, C. A. A. S. Metaheurística Simulated annealing para solução de problemas de planejamento florestal com restrições de integridade, *Revista Árvore*, 28, 247-256, 2004.

RODRIGUES, F. L.; SILVA, G. F.; LEITE, H. G.; XAVIER, A. C.; PEZZOPANE, J. E. M. Um modelo de regulação florestal e suas implicações na formulação e solução de problemas com restrições de recobrimento, *Revista Árvore*, 30, 769-778, 2006

SILVA, G. G. *Problemas no uso de programação matemática e Simulação em regulação florestal*, 89p. Tese (doutorado Em Ciência Florestal), Universidade Federal de Viçosa - UFV, 2001.

SILVA, G. F.; PIASSI, L. C.; MÔRA, R.; MARTINS, L. T.; TEIXEIRA, A. F. E.; BARROS JUNIOR, A. A. Metaheurística algoritmo genético na solução de modelos de planejamento florestal, *Agrária – Revista Brasileira de Ciências Agrárias*, 4, 160-166, 2009.

TEIXEIRA, A. F. *Aplicação de algoritmos evolucionários na solução de problemas de planejamento florestal multiobjectivo*, Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Minas Gerais - UFMG, 2002.

WEINTRAUB, A.; CHURCH, R. L.; MURRAY, A.; GUIGNARD, M. Forest management models and combinatorial algorithms: analysis of state of the art. *Operations Research*, 96, 271-285, 2000.