



Revista Árvore

ISSN: 0100-6762

r.arvore@ufv.br

Universidade Federal de Viçosa
Brasil

Rodrigues Lopes, Flávio; Leite Garcia, Helio; Santos Nascimento, Heleno do; Souza Lopes de,
Agostinho
Solução de problemas de planejamento florestal com restrições de inteireza utilizando busca tabu
Revista Árvore, vol. 27, núm. 5, setembro-outubro, 2003, pp. 701-713
Universidade Federal de Viçosa
Viçosa, Brasil

Disponível em: <http://www.redalyc.org/articulo.oa?id=48827513>

- Como citar este artigo
- Número completo
- Mais artigos
- Home da revista no Redalyc

redalyc.org

Sistema de Informação Científica
Rede de Revistas Científicas da América Latina, Caribe, Espanha e Portugal
Projeto acadêmico sem fins lucrativos desenvolvido no âmbito da iniciativa Acesso Aberto

SOLUÇÃO DE PROBLEMAS DE PLANEJAMENTO FLORESTAL COM RESTRIÇÕES DE INTEIREZA UTILIZANDO BUSCA TABU¹

Flávio Lopes Rodrigues², Helio Garcia Leite³, Heleno do Nascimento Santos⁴ e Agostinho Lopes de Souza³

RESUMO - Este trabalho teve como objetivos desenvolver e testar um algoritmo com base na metaheurística busca tabu (BT), para a solução de problemas de gerenciamento florestal com restrições de inteireza. Os problemas avaliados tinham entre 93 e 423 variáveis de decisão, sujeitos às restrições de singularidade, produção mínima e produção máxima periódicas. Todos os problemas tiveram como objetivo a maximização do valor presente líquido. O algoritmo para implementação da BT foi codificado em ambiente *delphi* 5.0 e os testes foram efetuados em um microcomputador AMD K6II 500 MHZ, com memória RAM de 64 MB e disco rígido de 15GB. O desempenho da BT foi avaliado de acordo com as medidas de eficácia e eficiência. Os diferentes valores ou categorias dos parâmetros da BT foram testados e comparados quanto aos seus efeitos na eficácia do algoritmo. A seleção da melhor configuração de parâmetros foi feita com o teste L&O, a 1% de probabilidade, e as análises através de estatísticas descritivas. A melhor configuração de parâmetros propiciou à BT eficácia média de 95,97%, valor mínimo igual a 90,39% e valor máximo igual a 98,84%, com um coeficiente de variação de 2,48% do ótimo matemático. Para o problema de maior porte, a eficiência da BT foi duas vezes superior à eficiência do algoritmo exato *branch and bound*, apresentando-se como uma abordagem muito atrativa para solução de importantes problemas de gerenciamento florestal.

Palavras-chave: Gerenciamento florestal, metaheurísticas e busca tabu.

SOLVING FOREST MANAGEMENT PROBLEMS WITH INTEGER CONSTRAINTS USING TABU SEARCH

ABSTRACT - This work aimed to develop and test an algorithm based on Tabu Search (TS) metaheuristics; to solve problems of forest management with integer constraints. TS was tested in five problems containing between 12 and 423 decision variables subjected to singularity constraints, minimum and maximum periodic productions. All the problems aimed at maximizing the net present value. TS was codified into *delphi* 5.0 language and the tests were performed in a microcomputer AMD K6II 500 MHZ, RAM memory 64 MB and hard disk 15GB. TS performance was evaluated according to efficacy and efficiency measures. The different values or categories for the TS parameters were tested and compared to their effects on algorithm efficacy. The best configuration of the parameters was selected by applying the L&O test at 1% probability and analyses via descriptive statistics. The best parameter configuration provided TS an average efficacy of 95.97%, a minimum value equal to 90.39%, a maximum value equal to 98.84%, with a variation coefficient of 2.48% of the mathematical optimum. For the most complex problem, TS efficiency of the exact algorithm *branch and bound*. Thus, TS is a rrather attractive approach to solving important forest management problems.

Key words: Forest Management, Heuristics and Tabu Search.

¹ Recebido para publicação em 16.5.2002.

Aceito para publicação em 9.9.2003.

² D.S. em Ciência Florestal, Dep. de Engenharia Florestal da Universidade Federal de Viçosa – UFV, 36570-000 Viçosa-MG;

³ Prof. do Dep. de Engenharia Florestal – UFV; ⁴ Prof. do Dep. de Informática – UFV.

1. INTRODUÇÃO

As metaheurísticas, um grupo especial das chamadas heurísticas estocásticas e analógicas (Goldberg & Luna, 2000), com aplicações em diversas áreas da ciência, têm sido ainda pouco exploradas no meio florestal. As principais metaheurísticas com grande potencial de aplicação na solução de problemas de planejamento florestal são: algoritmos genéticos (AG) (Holland, 1975; Goldberg, 1989), busca tabu (BT) (Glover, 1977) e *simulated annealing* (SA) (Kirkpatrick et al., 1983; Cerny, 1985). As facilidades de implementação e o grande sucesso dessas técnicas na solução de variados problemas são as suas principais vantagens em relação às heurísticas convencionais.

Dentre as metaheurísticas a BT tem sido bastante utilizada. Os princípios básicos dessa técnica têm sido usados de diferentes maneiras e com diversos nomes há aproximadamente 20 anos (Glazar, 2000). Ela teve início em procedimentos combinatórios em problemas de cobertura não-lineares no fim da década de 70 (Glover, 1977). As idéias básicas de sua forma atual foram desenvolvidas por Glover (1986) e Werra & Hertz (1989) e aperfeiçoada por Glover & Laguna (1997). A BT é aplicada a diversos tipos de problemas combinatórios. Os primeiros foram: seqüenciamento da produção, balanceamento de canais de comunicação e análise de agrupamentos. Recentemente ela passou a ser aplicada ao problema do caixeiro-viajante, coloração de grafos, projetos de circuitos integrados, roteamento de veículos, dentre outros (Glazar, 2000). Alguns exemplos de problemas de otimização combinatória no gerenciamento florestal incluem aqueles com restrições de singularidade, adjacência, roteamento de veículos no transporte

florestal, corte na indústria de móveis e de papel, dentre outros. Apesar do sucesso da BT na solução de problemas de diferentes áreas, existem poucos trabalhos que reportam a sua utilização na solução de problemas de gerenciamento florestal. Alguns exemplos são os trabalhos de Boston & Bettinger (1999), Laroze & Gleber (1997), Laroze (1999) e Richards et al. (2000) e Rodrigues (2001). Assim, idealizou-se este estudo com os seguintes objetivos:

- desenvolver e testar um algoritmo de busca tabu para resolver problemas de planejamento florestal;
- testar o efeito dos parâmetros da BT na estratégia de solução; e
- expandir a aplicação do algoritmo para problemas maiores e mais complexos, através do desenvolvimento de um código computacional para este fim.

2. MATERIAL E MÉTODOS

2.1. O Problema

Para testar o algoritmo BT desenvolvido neste trabalho, foram utilizados quatro problemas de planejamento florestal (Quadro 1). Os problemas diferenciam-se por variações no número de variáveis de decisão e número e magnitude das restrições de produção (demanda) mínimas e máximas.

Além das restrições de produção mínimas e máximas, todos os problemas tiveram como objetivos maximizar o valor presente líquido sujeito às restrições de singularidade (adotar um único regime de manejo em cada talhão).

Quadro 1 – Descrição geral dos problemas escolhidos para aplicação do AG, formulados através do sistema de planejamento florestal *SysFlor*

Table 1 – Overall description of the problemas selected for AG application, through the forest management system *SysFlor*

Problema	Nº Unidades de Manejo	Nº Variáveis de Decisão	Horizonte de Planejamento*	Produção Mínima (m³)	Produção Máxima (m³)
1	10	93	10	15.000,00	60.000,00
2	10	206	15	15.000,00	60.000,00
3	19	215	10	15.000,00	60.000,00
4	40	423	10	30.000,00	120.000,00

* Horizonte subdividido em períodos anuais.

2.2. O Modelo

Os problemas foram formulados mediante o uso do sistema de planejamento florestal *SysFlor* (Rodrigues et al., 2000). A abordagem de formulação utilizada pelo *SysFlor* segue a estrutura do modelo I proposta por Johnson & Scheurman (1977). A formulação matemática desse modelo para os problemas propostos está descrita a seguir.

Objetivo:

$$\text{Max } Z = \sum_{i=1}^M \sum_{j=1}^N c_{ij} x_{ij} \quad (1)$$

Sujeito a:

$$\sum_{i=1}^M \sum_{j=1}^N x_{ij} = 1 \quad (2)$$

$$\sum_{i=1}^M \sum_{j=1}^N V_{ijk} x_{ij} \geq Dmin_k \quad \{k = 0, 1, \dots, H - 1\} \quad (3)$$

$$\sum_{i=1}^M \sum_{j=1}^N V_{ijk} x_{ij} \leq Dmax_k \quad \{k = 0, 1, \dots, H - 1\} \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad (5)$$

em que Z = lucro total (\$); x_{ij} = variável de decisão representando a j -ésima alternativa de manejo adotada na i -ésima unidade de manejo; c_{ij} = valor presente líquido total (\$) de cada unidade de manejo i , colhida segundo a alternativa j ; M = número total de unidades de manejo; N = número total de alternativas de manejo na i -ésima unidade de manejo; V_{ijk} = volume (m^3) produzido pela i -ésima unidade de manejo quando a j -ésima alternativa de manejo é adotada, no início do período k ; e $Dmin_k$; $Dmax_k$ = demanda de madeira (m^3) mínima e máxima, respectivamente, em cada período (k) do horizonte de planejamento.

De acordo com esse modelo de programação inteira, a maximização do lucro (1) está sujeita às restrições de singularidade (2) e (5) e de produção mínima (3) e máxima (4) periodicamente. A imposição da restrição (5) define as variáveis de decisão na forma binária, o que obriga a escolha de uma única variável de decisão (alternativa de manejo) em cada unidade de manejo.

2.2. O Algoritmo BT

O algoritmo BT foi implementado no programa *SMAP-F* (Soluções Metaheurísticas Aplicadas ao

Planejamento Florestal), codificado em ambiente *delphi* 5.0. Os testes com o algoritmo foram efetuados em um microcomputador AMD K6II 500 MHZ, com memória RAM de 64 MB e disco rígido de 15GB. O esquema genérico do programa com o respectivo algoritmo está representado no fluxograma da Figura 1. Os detalhes desse algoritmo são descritos nos tópicos que se seguem.

2.2.1. Entrada de Dados

A entrada de dados da rotina para implementação da metaheurística BT é um arquivo no formato texto, contendo o modelo de programação inteira representativo do problema em foco, gerado pelo sistema de planejamento *SysFlor* (Rodrigues et al., 2000). Além do modelo para representar o problema a ser resolvido, a solução de um problema utilizando a metaheurística BT requer a definição dos valores dos seus diferentes parâmetros. Estes valores são definidos pelo usuário em um módulo específico, conforme discutido no item 2.2.10 deste trabalho.

2.2.2. Codificação da Solução

A representação das alternativas de manejo para um dado problema é feita com a utilização das variáveis de decisão X_{ij} , para representar a alternativa de manejo j ($j=1, 2, \dots, N$) assinalada à unidade de manejo i ($i=1, 2, \dots, M$). Um vetor $V(x) = \{X_{11}, X_{12}, \dots, X_{ij}\}$, para representar essas variáveis de decisão, foi esquematizado conforme Figura 2.

Uma solução (x) viável segundo a restrição de singularidade para o problema, formulada através do modelo de programação inteira com as variáveis de decisão na forma binária ($X_{ij} \in \{0, 1\}$) e avaliada segundo a função-objetivo $f(x)$, foi representada de acordo com o esquema da Figura 3.

2.2.3. Solução Inicial

A solução inicial (x_0) do algoritmo BT é gerada aleatoriamente. Com uma probabilidade uniformemente distribuída, uma variável de decisão X_{ij} é selecionada aleatoriamente em cada unidade de manejo, para receber valor igual a 1 (variável básica). As demais variáveis são assinaladas com valores iguais a 0 (variáveis não-básicas), garantindo a viabilidade da restrição de singularidade, ou seja, adotar uma única opção de manejo em cada talhão.

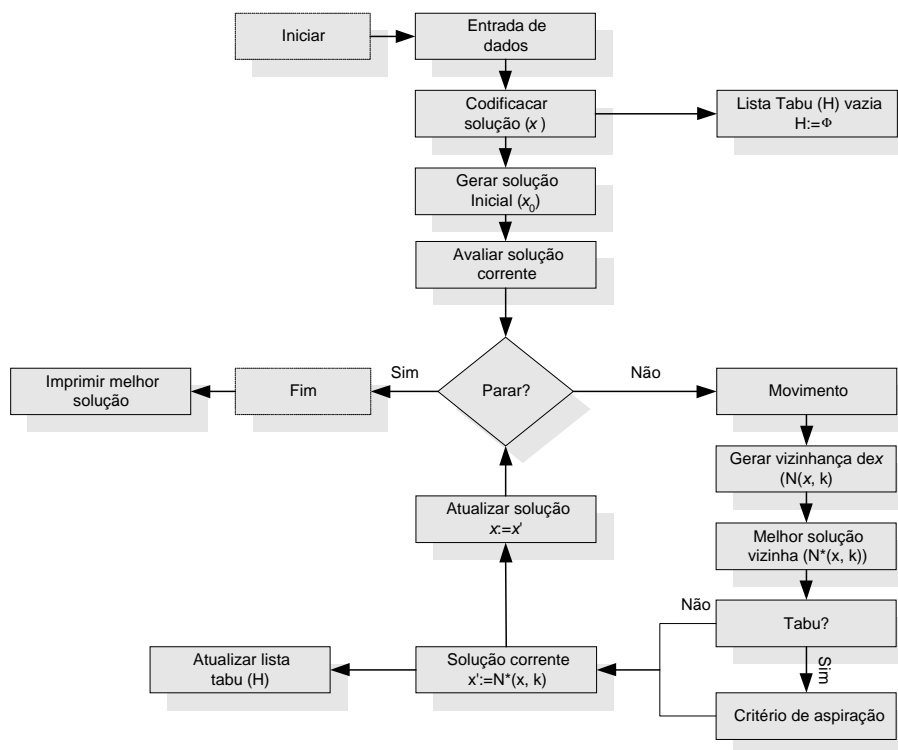


Figura 1 – SMAP-F: estrutura geral do programa com respectivo algoritmo para implementação da metaheurística busca tabu..

Figure 1 – SMAP-F: Overall program structure with respective algorithm for implementation of tabu search metaheuristics.



Figura 2 – Representação de um vetor com as variáveis de decisão do problema.

Figure 2 – Configuration of a vector with the problem-decision variables.



Figura 3 – Representação de uma solução para o problema com as variáveis de decisão na forma binária.

Figure 3 – Configuration for a problem solution, with binary decision variables.

2.2.4. Vizinhança e Movimentos

A BT opera sobre a noção de que uma vizinhança $N_i(x, k)$ pode ser construída através de um movimento m , para identificar soluções candidatas, em que x é uma

solução para o problema e k é a iteração corrente. Estes movimentos são utilizados para fugir de ótimos locais e devem ser hábeis para aceitar um movimento que leve de x para $x' \in N_i(x, k)$, mesmo quando $f(x') < f(x)$, em caso de problemas de maximização.

A vizinhança $N_i(x, k)$ é construída com o uso de um mecanismo de movimentos. A estratégia de movimento utilizada neste trabalho explora a vizinhança total ou parcial de uma dada solução x .

2.2.4.1. Vizinhança Total

Para gerar a vizinhança total de uma solução x , o mecanismo de movimento empregado, a partir da solução x , efetua todas as possíveis trocas de pares de variáveis (X_{ij}), transformando uma variável não-básica ($X_{ij} = 0$) em uma variável básica ($X_{ij} = 1$), e vice-versa. Para o problema de otimização discreta com as variáveis $X_{ij} \in \{0, 1\}$, a adição (*Adc*) de uma variável na base ($X_{ij} = 1$) implica a remoção (*Rem*) de uma outra variável da base ($X_{ij} = 0$). Utilizando, por exemplo, um problema fictício, com quatro unidades de manejo e 12 variáveis de decisão, uma vizinhança de x para esse problema seria a apresentada no Quadro 2. A cada k iterações são geradas $N_i(x, k)$ soluções vizinhas de x , em que $N_i(x, k)$ é a vizinhança completa de x , obtida pelo mecanismo de movimento determinístico de adição e remoção de pares de variáveis em cada unidade de manejo (Quadro 2). A vizinhança total (Vt) de uma dada solução x pode ser calculada por:

$$Vt = \sum_{i=1}^M (N_i - 1) \quad (6)$$

em que Vt = vizinhança total; M = número de unidades de manejo do problema; e N_i = número de alternativas de manejo em cada unidade de manejo.

O mecanismo apresentado anteriormente possui a vantagem de, partindo-se de uma solução x , poder gerar a sua vizinhança total, o que constitui uma estratégia de busca muito eficiente, já que toda vizinhança de x é avaliada. Entretanto, para problemas maiores, o número de soluções vizinhas pode ser muito elevado, depreciando o tempo computacional para convergência do algoritmo. Por exemplo, utilizando a expressão (6) em um problema com apenas 20 unidades de manejo, cada uma contendo 11 alternativas de manejo, seria necessário gerar e avaliar 200 soluções vizinhas. Uma alternativa para diminuir o número de soluções vizinhas é restringir o tamanho da vizinhança. Neste caso, apenas uma vizinhança parcial de x deve ser avaliada.

2.2.4.2. Vizinhança Parcial

Na vizinhança parcial (V_p), apenas uma porção da vizinhança total (Vt) de x é avaliada em cada iteração. O desafio agora passa a ser determinar quais soluções vizinhas serão avaliadas. O mecanismo empregado neste trabalho consiste em escolher esta vizinhança aleatoriamente. Neste caso, a escolha da nova variável básica ($X_{ij} = 1$) em cada unidade de manejo é feita de forma aleatória. A variável candidata a deixar a base ($X_{ij} = 0$) em cada unidade de manejo é aquela cujo valor é igual a 1 na solução x . No Quadro 3 está uma vizinhança parcial de x para um problema fictício, com quatro unidades de manejo e 12 variáveis de decisão geradas através do mecanismo de movimento descrito anteriormente. Considerou-se uma vizinhança parcial (N_v) igual a 5.

Quadro 2 - Vizinhança total de x gerada através do mecanismo de adicionar e remover deterministicamente pares de variáveis X_{ij} em cada unidade de manejo i

Table 2 - Total neighbourhood of x generated by deterministically adding and removing X_{ij} variable pairs in each i management unit

	UM (i)	1			2			3			4			Movimento	
	Variáveis	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}	X_{41}	X_{42}	X_{43}	Adc	Rem
	x	0	1	0	1	0	0	0	0	1	1	0	0	-	-
Vizinhança - $N_i(x, k)$	$N_1(x, k)$	1	0	0	1	0	0	0	0	1	1	0	0	X_{11}	X_{12}
	$N_2(x, k)$	0	0	1	1	0	0	0	0	1	1	0	0	X_{13}	X_{12}
	$N_3(x, k)$	0	1	0	0	1	0	0	0	1	1	0	0	X_{22}	X_{21}
	$N_4(x, k)$	0	1	0	0	0	1	0	0	1	1	0	0	X_{23}	X_{21}
	$N_5(x, k)$	0	1	0	1	0	0	1	0	0	1	0	0	X_{31}	X_{33}
	$N_6(x, k)$	0	1	0	1	0	0	0	1	0	1	0	0	X_{32}	X_{33}
	$N_7(x, k)$	0	1	0	1	0	0	0	1	0	0	1	0	X_{42}	X_{41}
	$N_8(x, k)$	0	1	0	1	0	0	0	1	0	0	0	1	X_{43}	X_{41}

Quadro 3 – Vizinhaça parcial de x de tamanho igual a 5, gerada através do mecanismo de adicionar e remover aleatoriamente pares de variáveis X_{is} em cada unidade de manejo (UM)

Table 3 – Partial neighbourhood of x of size equal to 5, generated by randomly adding and removing X_{is} variable pairs in each management unit (MU)

	UM (i)	1			2			3			4			Movimento	
	Variáveis	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}	X_{41}	X_{42}	X_{43}	Adc	Rem
	x	0	1	0	1	0	0	0	0	1	1	0	0	-	-
Vizinhaça – $N_i(x, k)$	$N_1(x, k)$	0	1	0	0	1	0	0	0	1	1	0	0	X_{22}	X_{21}
	$N_2(x, k)$	0	1	0	1	0	0	0	0	1	0	0	1	X_{43}	X_{41}
	$N_3(x, k)$	1	0	0	1	0	0	0	0	1	1	0	0	X_{11}	X_{12}
	$N_4(x, k)$	0	1	0	0	0	1	0	0	1	1	0	0	X_{23}	X_{21}
	$N_5(x, k)$	0	1	0	1	0	0	0	1	0	1	0	0	X_{32}	X_{33}

2.2.5. Função de Avaliação

O mecanismo de movimento utilizado neste trabalho gera soluções viáveis segundo as restrições de singularidade. Em relação às demais restrições (produção mínima e máxima), não há nenhuma garantia de viabilidade das soluções candidatas geradas. Em vez de testar a viabilidade dessas soluções candidatas em relação às demais restrições do problema e aceitar apenas soluções viáveis segundo essas restrições, o mecanismo de avaliação utilizado pela metaheurística BT utiliza uma função de penalidade para guiar a escolha de soluções candidatas. A função de penalidade, $f_p(x)$, é obtida através de alterações na função-objetivo, $f(x)$, do problema, da seguinte forma:

$$f_p(x) = f(x) - vp \times VT \quad (7)$$

em que $f_p(x)$ = valor da função-objetivo penalizada para a solução x ; $f(x)$ = valor da função-objetivo do problema; vp = penalização (\$/m³) para cada unidade de produção violada; VT = violação total (m³) das restrições de produção (mínima e máxima), conforme definido por Rodrigues (2001); e demais variáveis já definidas.

De acordo com a expressão (6), quanto maior o valor da violação total (VT) da solução candidata, maior será a penalidade atribuída a esta solução e menor será o seu desempenho em relação à função-objetivo penalizada ($f_p(x)$), o que contribui para desencorajar a escolha dessa solução. Quando a violação total é nula ($VT=0$), os valores de $f_p(x)$ e $f(x)$ se equivalem.

2.2.6. Memória de Curto Prazo e Restrições Tabu

O objetivo da restrição tabu é proibir movimentos que levem a soluções recentemente visitadas, utilizando algum mecanismo de controle. A BT lida com este problema, incorporando uma estrutura de memória que proíbe ou penaliza certos movimentos de retornar a uma solução recentemente visitada. Os movimentos proibidos são chamados de *movimento tabu*. O intervalo definido de k iterações no qual um movimento permanece tabu é denominado de *permanência* ($|T|$). Esta técnica de memória é conhecida como memória de curto prazo, uma vez que contém as soluções mais recentemente visitadas, prevenindo ciclos menores ou iguais a $|T|$.

Há várias formas de incorporar uma restrição tabu. Para o problema em foco, devido à natureza das restrições de singularidade (assinalamento de um único regime de manejo, X_{ij} , em cada talhão), a imposição de restrição tabu torna-se relativamente simples.

Seja X_{ij}^{Adc} a variável adicionada na melhor solução, $N_i^*(x, k)$, a restrição tabu imposta proíbe que X_{ij}^{Adc} seja removida (*Rem*) durante um período $|T|$ de iterações. Portanto, os movimentos tabu são todos aqueles que resultarem na remoção de X_{ij}^{Adc} durante $|T|$ iterações. Na prática, o controle da restrição tabu pode ser feito para cada unidade de manejo, uma vez que proibir uma variável de decisão X_{ij}^{Adc} de ser removida nas próximas k iterações ($|T|$) significa proibir qualquer movimento que resulte da troca de variáveis na respectiva unidade de manejo, pois a restrição de singularidade atribui a cada

unidade de manejo uma única variável de decisão, com valor igual a 1. Portanto, o número de soluções candidatas em um dado instante depende do número de unidades de manejo não-tabu. Seja M_{NT} o número de unidades de manejo classificadas como não-tabu em uma dada iteração, o tamanho da vizinhança em uma dada iteração é função dos movimentos derivados das unidades de manejo classificadas como não-tabu, assim:

$$N_v = \sum_{i=1}^{M_{NT}} (N_{iNT} - 1) \quad (8)$$

em que N_v = tamanho da vizinhança em um dado instante; M_{NT} = número de unidades de manejo não-tabu; e N_{iNT} = número de alternativas de manejo na i -ésima unidade de manejo não-tabu.

O tamanho da vizinhança (N_v) é máximo ($N_v = V_t$) somente quando todos os movimentos são classificados como não-tabu, o que em geral ocorre na primeira iteração. Se todas as unidades de manejo forem classificadas como tabu (neste caso, $M_{NT} = 0$), não haverá nenhum movimento disponível e, portanto, o tamanho da vizinhança será igual a zero.

2.2.7. Lista Tabu e Atributos do Movimento

O tipo da estrutura das listas tabu depende do problema em questão, e não existe nenhuma estrutura que seja melhor para todas as aplicações. Entretanto, algumas regras gerais devem ser levadas em consideração no momento em que se planeja a estrutura da lista. A grande

preocupação em encontrar a melhor estrutura de lista deve-se a fatores como a eficiência computacional, o que reforça a tese de que cada problema deve ter sua estrutura de lista definida com base na experimentação, quando na ausência de indicadores de uma estrutura mais adequada. A lista tabu utilizada pelo algoritmo BT implementado neste trabalho armazena o intervalo de iterações ou as próximas iterações em que o movimento permanecerá como movimento tabu (Quadro 4).

Apesar de não serem armazenados na lista tabu propriamente dita, alguns atributos do movimento como a variável adicionada (Adc), a variável removida (Rem), a configuração da solução (x), as soluções vizinhanças ($N_i(x, k)$), o valor da função objetivo penalizada ($fp_i(x)$), a função-objetivo não-penalizada ($f_i(x)$), a violação total da i -ésima solução (VT_i), dentre outros, são aspectos que conferem informações à BT (Quadro 4). A forma de cálculos, de armazenamento e de recuperação desses atributos pode ainda afetar o desempenho da BT.

A permanência tabu de um movimento define o tamanho da lista tabu ($|T|$). O tamanho da lista pode ser estático, sem alteração durante toda a busca, ou dinâmico, variando de tamanho ao longo do processo de busca. Apesar de haver algumas regras práticas para determinação de $|T|$, existe um consenso na literatura, reportando sobre BT, de que a definição do tamanho de $|T|$ e de sua forma de operação (estática ou dinâmica) é ainda dependente de experimentação. Neste trabalho foram empregadas duas formas de permanência tabu, ou seja, a estática ($|T_e|$) e a dinâmica ($|T_d|$).

Quadro 4 – Lista tabu representando as próximas k iterações com o melhor movimento escolhido considerado tabu e alguns atributos do movimento da metaheurística BT

Table 4 – Tabu search list representing the next k iterations with the best chosen movement considered tabu and some T_s metaheuristic movement attributes

Iteração	Solução	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}	X_{41}	X_{42}	X_{43}	Adc	Rem	$f_i(x)$	$fp_i(x)$	$VT_i(x)$	$ T $
k	x	0	0	1	0	0	1	0	1	0	0	0	1	-	-	-	-	-	-
Vizinhança - $N_i(x, k)$	$N_1(x, k)$	0	1	0	0	0	1	0	1	0	0	0	1	X_{12}	X_{13}	$f_1(x)$	$fp_1(x)$	$VT_1(x)$	3
	$N_2(x, k)$	1	0	0	0	0	1	0	1	0	0	0	1	X_{11}	X_{13}	$f_2(x)$	$fp_2(x)$	$VT_2(x)$	3
	$N_3(x, k)$	0	0	1	0	1	0	0	1	0	0	0	1	X_{22}	X_{23}	0
	$N_4(x, k)$	0	0	1	1	0	0	0	1	0	0	0	1	X_{21}	X_{23}	0
	$N_5(x, k)$	0	0	1	0	0	1	1	0	0	0	0	1	X_{31}	X_{32}				0
	$N_6(x, k)$	0	0	1	0	0	1	0	0	1	0	0	1	X_{33}	X_{32}				0
	$N_7(x, k)$	0	0	1	0	0	1	0	1	0	1	0	0	X_{42}	X_{43}	2
	$N_8(x, k)$	0	0	1	0	0	1	0	1	0	0	1	0	X_{41}	X_{43}	$f_8(x)$	$fp_8(x)$	$VT_8(x)$	2

No tamanho estático o valor da permanência ($|Te|$) escolhido fica inalterado durante toda a busca.

No tamanho dinâmico ($|TD|$), o valor da permanência tabu varia ao longo da busca de acordo com a função de distribuição uniforme, dentro de um intervalo de variação mínimo (t_{min}) e máximo (t_{max}), ou seja, $|TD| \in \text{uniforme}\{t_{min}, t_{max}\}$. O valor de $|TD|$ escolhido em uma dada iteração pode permanecer constante durante α iterações, quando um novo valor de permanência é selecionado pelo mesmo processo. A determinação dos extremos t_{min} e t_{max} da distribuição uniforme é importante para propiciar uma duração tabu balanceada, evitando ciclagem e degeneração da busca.

2.2.8. Critérios de Aspiração

Pode ocorrer no processo de busca que a vizinhança, em um dado instante, seja vazia, isto é, que todos os movimentos candidatos sejam considerados tabu, caso em que a busca seria interrompida. Para contornar esta situação pode-se dominar o estado tabu dos movimentos para cancelá-los quando necessário e permitir que a busca prossiga. A BT implementada neste trabalho permite o cancelamento do estado tabu de um movimento, e desta forma libera a sua execução, permitindo o prosseguimento da busca, utilizando-se os critérios de aspiração *default* e por objetivo. Porém, antes de aplicar os critérios de aspiração, o mecanismo de busca necessita gerar uma nova vizinhança. Para tal, escolheu-se uma nova vizinhança tabu parcial (VP) ou total (Vt) para eleger, dentre os movimentos desta vizinhança, aquele que terá o seu estado tabu cancelado. Se nenhum desses movimentos melhorar o valor da melhor solução até então encontrada, o critério de aspiração *default* é usado para cancelar o estado tabu de um movimento com a condição tabu mais antiga. Havendo empate, o movimento escolhido será aquele que possuir melhor valor da função objetivo. No critério de aspiração por objetivo, o estado tabu do movimento cancelado é aquele que corresponde ao movimento da vizinhança tabu com o valor da função-objetivo melhor que o valor da função-objetivo da melhor solução até então encontrada.

2.2.9. Critério de Parada

O algoritmo BT proposto utiliza como critério de parada a primeira solução viável (violação total nula, $VT=0$), mais um número de iterações adicionais (Δk), para tentar melhorar a solução viável encontrada. Assim,

a BT tenta encontrar várias soluções viáveis, escolhendo aquela de melhor valor para a função-objetivo ($f^*(x)$). Para interromper a corrida da BT após um número de iterações sem encontrar uma solução viável para o problema, foi definido um número máximo de iterações (k_{max}).

2.2.10. Escolha dos Parâmetros da BT

Os parâmetros da metaheurística busca tabu avaliados neste trabalho foram os tipos de permanência (estática e dinâmica), o tamanho da permanência estática e o tamanho da vizinhança. Conforme já demonstrado, a permanência dinâmica é função dos parâmetros t_{min} , t_{max} e α . Além destes parâmetros básicos, a metaheurística BT utiliza outros parâmetros como: o valor da penalização (vp), para cálculos das penalidades; o número de interações adicionais (Δk), para melhorar a qualidade das soluções viáveis encontradas; e o número máximo de iterações permitidas (k_{max}).

Um resumo dos parâmetros e seus respectivos valores utilizados no teste da BT estão apresentados no Quadro 5. As combinações entre todos os níveis dos diferentes parâmetros gerou um ensaio com 32 diferentes combinações, representando um cenário ou tratamento diferente. Cada tratamento foi repetido quatro vezes, resultando em 128 observações para cada problema ou um total de 512 observações.

Os valores de permanência estática e o limite superior da permanência dinâmica foram fixados como uma função do número de unidades de manejo (M), variável de acordo com o problema avaliado. Da mesma forma, o tamanho da vizinhança parcial é função da vizinhança total (Vt), que também varia de acordo com o problema.

2.3. Avaliação do Desempenho da BT

O desempenho da BT foi avaliada através da sua eficácia, calculada pela razão expressa em porcentagem entre o melhor valor da função-objetivo obtida em uma corrida da BT e o ótimo matemático obtido pelo algoritmo exato *branch and bound*, assim:

$$Ef = \frac{f_{BT}}{f_o} \times 100 \quad (9)$$

em que Ef = eficácia (%); f_{BT} = valor da melhor solução obtida em uma corrida da BT (\$); e f_o = valor da solução ótima obtida pelo algoritmo *branch and bound* (\$).

Quadro 5 – Resumo dos parâmetros utilizados pela metaheurística busca tabu**Table 5** – Summary of the parameter used by tabu search metaheuristics

Parâmetro	Nível	Valor ou Categoria dos Parâmetros			
Tipo de permanência	2	Estática	Dinâmica		
Permanência estática ($ Te $)	4	0,25M	0,50M	0,75M	M
Tamanho máximo da vizinhança parcial (VP)	4	0,10Vt	0,20Vt	0,30Vt	0,40Vt
Limite mínimo para permanência dinâmica (t_{min})	1	1			
Limite máximo para permanência dinâmica (t_{max})	1	M			
Iterações para troca da permanência dinâmica (α)	1	1			
Número máximo de interação (k_{max})	1	300			
Iterações adicionais (Δk)	1	50			

Em que: M = número de unidades de manejo do problema e Vt é a vizinhança total do problema.

As medidas adequadas de eficiência dependem de estudos mais sofisticados, como a análise de complexidade de algoritmos, tais como complexidade de tempo e de espaço. Neste trabalho, a eficiência da BT foi avaliada utilizando-se somente o tempo de processamento do algoritmo, o que permite apenas comparações relativas entre os métodos de solução, processadas com os mesmos recursos computacionais. As comparações entre os valores dos diferentes parâmetros foram efetuadas com o procedimento estatístico, denominado teste L&O, proposto por Leite & Oliveira (2002).

3. RESULTADOS E DISCUSSÃO

As medidas de eficácia da BT para os problemas avaliados, considerando diferentes valores de permanência estática, estão apresentadas na Figura 4. Apenas os valores de permanência estática iguais a 0,25 e 0,50M não diferiram entre si pelo teste de L&O, a 1% de probabilidade. Estes valores foram também os que propiciaram maior eficácia na BT, quando comparados aos valores correspondentes a 0,75M e M (Figura 4). Por exemplo, para os problemas 1 e 2, contendo dez unidades de manejo, os valores de permanência foram fixados como sendo 3, 5, 8 e 10. Cabe ressaltar que os valores 3 e 8 foram obtidos por arredondamento dos valores fracionados 2,5 (0,25M) e 7,5 (0,75 M), respectivamente. A permanência é um dos principais parâmetros da BT, podendo afetar o seu desempenho. Valores baixos podem diminuir a diversidade da busca devido à ciclagem e degeneração, podendo levar ao estacionamento em ótimos locais pela dificuldade de exploração de novas soluções. Valores altos, apesar de favorecem a pesquisa

de novas soluções, funcionando como um artifício para fuga de ótimos locais, podem também diminuir a diversidade da busca devido à diminuição no tamanho da vizinhança, reduzindo as chances de encontrar uma solução de melhor qualidade. Este argumento parece ter sido o que mais afetou o desempenho da BT, ou seja, maiores valores de permanência resultaram em menor eficácia da BT devido à redução da vizinhança, com conseqüente reflexo no desempenho da BT.

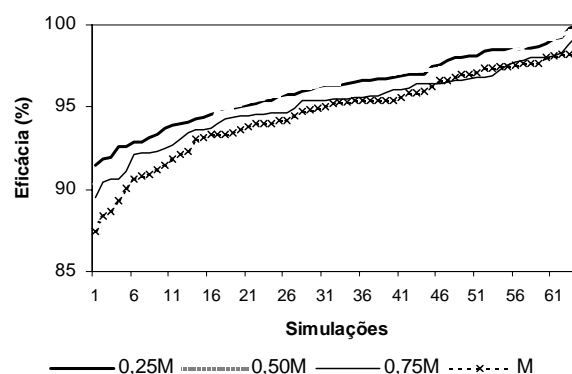


Figura 4 – Medidas de eficácia para diferentes valores de permanência estática da metaheurística BT nos quatro problemas analisados.

Figure 4 – Efficacy measurements for different values of static permanence of TS metaheuristics for the four problems analyzed.

Não houve diferença estatística significativa, pelo teste de L&O a 1% de probabilidade, entre a permanência dinâmica, comparada aos melhores valores de permanência estática (0,25M e 0,50M) na eficácia da BT

(Figura 5). Cabe ressaltar que a permanência dinâmica adotada neste trabalho considerou o intervalo de variação com limite mínimo igual a 1 ($t_{min}=1$), limite máximo igual ao número de unidades de manejo do problema ($t_{max}=M$) e a escolha de um novo valor de permanência a cada nova iteração ($\alpha=1$). Como não houve diferenças estatísticas significativas entre a permanência dinâmica e as melhores configurações de permanência estática, pode-se inferir que nas situações em que não haja nenhum indicativo do melhor valor de permanência estática a permanência dinâmica possa ser adotada, principalmente devido à maior flexibilidade na definição deste parâmetro. É necessário investigar o efeito da permanência dinâmica para outros valores de t_{min} , t_{max} e α , principalmente em virtude da possibilidade de melhorias na eficácia da BT para tais combinações.

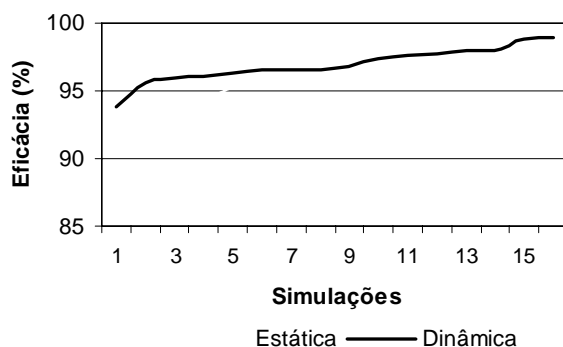


Figura 5 – Medidas de eficácia para dois tipos de permanência da metaheurística BT nos quatro problemas analisados.

Figure 5 – Efficacy measurements for two types of TS metaheuristics permanence in the four problems analyzed.

Em relação ao tamanho da vizinhança parcial máxima, apenas o tamanho igual a $0,1Vt$ diferiu dos demais valores pelo teste de L&O, a 1% de probabilidade (Figura 6). Maiores valores de vizinhança parcial máxima propiciaram maiores valores de eficácia da BT.

O tamanho da vizinhança parcial máxima varia com o da vizinhança total (Vt) de cada problema. No Quadro 6 está apresentada a vizinhança total dos quatro problemas avaliados. Os valores apresentados dependem do tipo de movimento empregado (neste trabalho, eles foram derivados por mecanismo de adicionar e remover

em cada unidade de manejo pares de variáveis de decisão, X_{ij}). Outra observação importante é que a vizinhança total, em um dado instante, depende do número de unidades de manejo classificadas como não-tabu. Portanto, a vizinhança total é máxima quando todas as unidades de manejo forem não-tabu, o que em geral ocorre na primeira iteração. O tamanho da vizinhança é nulo quando todas as unidades de manejo são classificadas como tabu, quando a busca seria interrompida devido à indisponibilidade de novos movimentos.

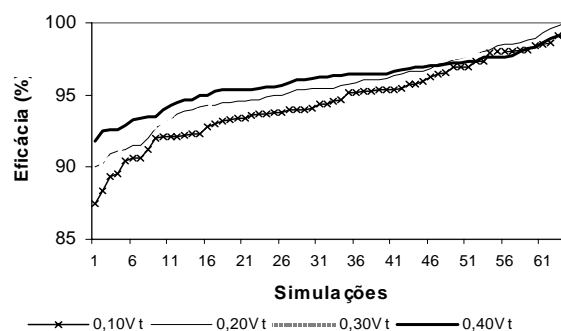


Figura 6 – Medidas de eficácia para diferentes tamanhos de vizinhanças da metaheurística BT nos quatro problemas analisados.

Figure 6 – Efficacy measurements for different TS metaheuristics neighbourhood sizes in the four problems analyzed.

Quadro 6 – Vizinhança total de cada problema, considerando o mecanismo de movimento de adicionar e remover em cada unidade de manejo pares de variáveis de decisão

Table 6 – Total neighbourhood of each problem, based on the mechanism of adding and removing decision variable pairs in each unit

Problema	Vizinhança Total (Vt)
1	83
2	196
3	196
4	384

Para evitar que a busca seja interrompida pela falta de movimentos candidatos, a BT permite o cancelamento do estado tabu de alguns movimentos para permitir o prosseguimento da busca. O mecanismo que permite o

cancelamento do estado tabu dos movimentos é o critério de aspiração. Neste trabalho foram utilizados dois critérios de aspiração: *default* e por objetivo. O critério de aspiração *default* permitiu o cancelamento do estado tabu do movimento com a condição tabu mais antiga, naquelas situações em que nenhum movimento candidato melhorava o valor da melhor solução até então encontrada. Por outro lado, o critério de aspiração por objetivo permitiu a escolha de um novo movimento, se este resultasse em uma melhoria da função-objetivo da melhor solução até então encontrada. Na estratégia de busca empregada, os dois critérios são mutuamente exclusivos, ou seja, a escolha do critério empregado depende da avaliação de uma nova vizinhança gerada pelo cancelamento temporário do estado tabu de todos os movimentos disponíveis. O tamanho da vizinhança neste instante é igual ao valor da vizinhança parcial máxima escolhido para uma dada corrida.

Ao selecionar apenas os melhores valores de permanência estática (0,25M e 0,50M) e os melhores tamanhos de vizinhança (0,30Vt e 0,40Vt), a BT apresentou eficácia mínima, média e máxima iguais a 90,39, 95,97 e 98,84%, respectivamente (Figura 7). Neste caso, o coeficiente de variação foi igual a 2,08%.

A principal desvantagem de se ter resultados melhores de eficácia da BT com maiores valores de vizinhanças deve-se à depreciação no tempo computacional da busca com o aumento do tamanho da vizinhança (Quadro 7). Por exemplo, para o problema de maior porte

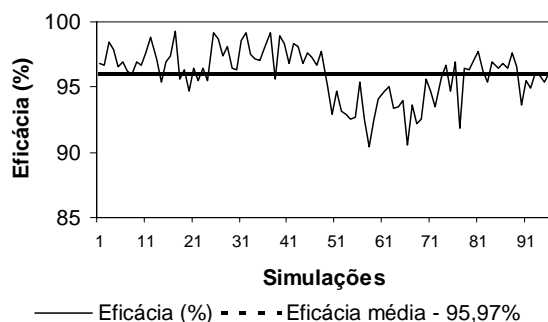


Figura 7 – Medidas de eficácia da BT para os quatro problemas analisados, considerando os melhores tamanhos de permanência estática (0,25M e 0,50M) e os melhores tamanhos de vizinhança (0,30Vt e 0,40Vt).

Figure 7 – TS efficacy measurements for the four problems analyzed, based on the best static permanence sizes (0.25M e 0.50M) and the best sizes (0.30Vt e 0.40Vt).

(problema 4), ao se quadruplicar o tamanho da vizinhança, houve um aumento na mesma proporção no tempo computacional. Este aumento deve-se ao maior esforço exigido pela BT para gerar e avaliar um maior número de soluções candidatas com o aumento da vizinhança.

Quadro 7 – Tempos médios (segundos) de processamento da BT para diferentes valores da vizinhança parcial máxima para os quatro problemas avaliados

Table 7 – Mean times (seconds) of TS processing for different maximum partial neighbourhood values for the four problems analyzed

Problema	Tamanho da Vizinhança Parcial Máxima			
	10% Vt	20% Vt	30% Vt	40% Vt
1		2	4	6
2	21	34	50	64
3	20	40	75	79
4	170	245	466	599

Em que Vt é a vizinhança total não-tabu.

A eficiência da BT não foi avaliada rigorosamente neste trabalho. Entretanto, os resultados computacionais preliminares da BT mostraram a superioridade desse método quando comparado ao algoritmo exato *branch and bound*. O tempo computacional gasto pelo algoritmo *branch and bound* cresce mais rapidamente com o tamanho do problema (número de variáveis), comparado à BT. No problema 4, com 423 variáveis, o algoritmo *branch and bound* consumiu um tempo duas vezes superior ao da BT (Figura 8).

Apesar das vantagens já obtidas, o tempo de processamento da BT deve ser melhorado. Melhorias na estrutura de armazenamento e recuperação de dados, mecanismos de cálculos mais eficientes para avaliações das restrições e função-objetivo do problema e mecanismos para geração de movimentos e classificação tabu são apenas alguns dos aspectos da BT que podem ser conseguidos. Tais melhorias são vitais para aumentar a capacidade de processamento da BT em tempos computacionais viáveis, e assim viabilizar as suas aplicações em problemas de maior porte.

Outro aspecto que pode melhorar ainda mais o desempenho da BT, principalmente em relação à melhoria da sua eficácia, é a incorporação de outros artifícios da BT, como a inclusão de estruturas de memória de longo

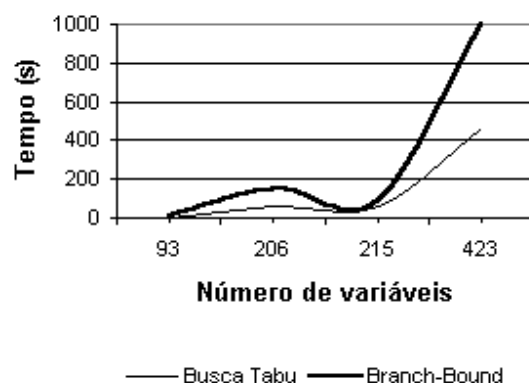


Figura 6 – Tempo de solução obtido pela metaheurística busca tabu e o algoritmo *branch and bound*.

Figure 6 – Solution time obtained by tabu search metaheuristics and the algorithm *branch and bound*.

prazo, aspecto ainda não explorado neste trabalho. A inclusão dessas estruturas de memória podem ser decisivas para o sucesso da BT em problemas maiores e, ou, mais complexos. Além disto, outros fatores como a qualidade do critério para interrupção da busca podem afetar o desempenho da BT e necessitam ser mais bem investigados.

4. CONCLUSÕES

Este trabalho utilizou a metaheurística busca tabu (BT) para resolver problemas de planejamento florestal com restrições de inteireza. Foram testados quatro problemas, contendo entre 93 e 423 variáveis de decisão, sujeitos às restrições de singularidade e produção mínima e máxima, com o objetivo de maximizar a receita líquida.

Algumas configurações de parâmetros propiciaram melhor eficácia para a BT. Neste sentido, os valores de permanência estática fixada em 25 e 50% do número de unidade de manejo do problema apresentaram eficácias superiores aos demais valores (75 e 100%). A permanência dinâmica foi uma boa estratégia para melhorar a eficácia da BT, pois apresentou resultados semelhantes aos melhores valores de permanência estática. Tamanhos maiores de vizinhança parcial (30 e 40% da vizinhança total) propiciaram maiores eficácias para a BT. Entretanto a eficiência da BT, avaliada pelo tempo gasto para encontrar a solução do problema, sofreu uma depreciação para os maiores tamanhos de vizinhança.

Para os quatros problemas estudados, os melhores parâmetros da BT forneceram soluções com eficácia média de 95,97%, valor mínimo igual a 90,39%, valor máximo igual a 98,84% do ótimo matemático, com um coeficiente de variação igual a 20,08%. O tempo de processamento para encontrar uma boa solução para o problema de maior porte, considerando uma vizinhança parcial de 30% da vizinhança total, foi duas vezes inferior ao principal algoritmo exato de solução (*branch and bound*).

A incorporação de outros artifícios da BT como a inclusão de estruturas de memória de longo prazo, aspecto ainda não explorado neste trabalho, pode contribuir ainda mais para o sucesso da BT em resolver problemas de planejamento florestal, principalmente os de grande porte e, ou, de maior complexidade.

A aplicabilidade universal da BT, propiciada pela flexibilidade desses algoritmos em resolver problemas de diferentes domínios, cabendo apenas adaptá-los aos seus padrões, permite inferir que esta técnica pode ser utilizada para resolver uma variedade de outros importantes problemas de planejamento florestal, entre eles aqueles que envolvem restrições de adjacência, roteamento de veículos para o transporte florestal, corte na indústria seccionamento de toras etc.

Os objetivos deste trabalho foram alcançados, porém é necessário melhorar a capacidade da rotina computacional para viabilizar a aplicação da busca tabu em problemas maiores.

5. AGRADECIMENTO

À Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

6. REFERÊNCIAS BIBLIOGRÁFICAS

BOSTON, K.; BETTINGER, P. An Analysis of Monte Carlo Integer Programming, Simulated Annealing, and Tabu Search Heuristics for Solving Spatial Harvest Scheduling Problems. **Forest Science**, v. 45, n. 2, p. 292-301, 1999.

CERNY, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. **Journal of Optimization Theory and Application**, n. 45, n. 1, p. 41-51, 1985.

- GLAZAR, E. J. N. **Análise de diferentes estratégias na aplicação da busca tabu ao problema de roteamento de veículos**. 2000. 132 f. Dissertação (Mestrado em Ciência em Engenharia de Produção) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2000.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers and Operational Research**, v. 5, p. 533-549, 1986.
- GLOVER, F. Heuristics for integer programming using surrogate constraints. **Decision Science**, v. 8, p. 156-166, 1977.
- GLOVER, F. Heuristics for integer programming using surrogate constraints. **Decision Science**, v. 8, p. 156-166, 1977.
- GLOVER, F.; LAGUNA, M. **Tabu search**. Boston: Kluwer Academic Publishers, 1997. 382 p.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Campus, 2000. 649 p.
- GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Addison Wesley, 1989.
- HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Harbor: University of Michigan Press, 1975.
- KIRKPATRICK JR., S.; GELATT, C.; VECCHI, M. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 498-516, 1983.
- LAROZE, A. J. A linear programming, tabu search method for solving forest-level bucking optimization problems. **Forest Science**, v. 45, n.1, p. 108-116, 1999.
- LAROZE, A. J.; GREBER, B. J. Using Tabu search to generate stand-level, rule-based bucking patterns. **Forest Science**, v. 43, n. 2, p. 157-169, 1997.
- LEITE, H. G.; OLIVEIRA, F. H. T. Statistical procedure to test the identity of analytical methods. **Communications in soil science and plant analysis**. New York: 2002. p. 1105-1118.
- RICHARDS, E. W.; GUNN, E. A. A Model and tabu search method to optimize stand harvest and road construction schedules. **Forest Science**, v. 46, n. 2, p. 188-203, 2000.
- RODRIGUES, F. L. **Metaheurística e sistema de suporte à decisão no gerenciamento de recursos florestais**. 2001. 225 f. Tese (Doutorado em Manejo Florestal) – Universidade Federal de Viçosa, Viçosa, 2001.
- RODRIGUES, F. L.; LEITE, H. G.; ALVES, J. M. *Sysflor* – um sistema de suporte à decisão para o planejamento florestal. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 32., 2000, Viçosa. **Anais...** Viçosa: UFV, 2000. p. 974-995.
- WERRA, D.; HERTZ, A. Tabu search techniques: a tutorial and an application to neural networks. **OR Spektrum**, v. 11, p. 131-141, 1989.