

METAHEURÍSTICA *SIMULATED ANNEALING* PARA SOLUÇÃO DE PROBLEMAS DE PLANEJAMENTO FLORESTAL COM RESTRIÇÕES DE INTEGRIDADE¹

Flávio Lopes Rodrigues², Helio Garcia Leite³, Heleno do Nascimento Santos⁴, Agostinho Lopes de Souza³ e Carlos Antônio Álvares Soares Ribeiro³

RESUMO - Os objetivos deste trabalho foram desenvolver e testar a metaheurística SA para solução de problemas de gerenciamento florestal com restrições de integridade. O algoritmo SA desenvolvido foi testado em quatro problemas, contendo entre 93 e 423 variáveis de decisão, sujeitos às restrições de singularidade, produção mínima e produção máxima, periodicamente. Todos os problemas tiveram como objetivo a maximização do valor presente líquido. O algoritmo SA foi codificado em linguagem *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 SA foi avaliado de acordo com as medidas de eficácia e eficiência. Os diferentes valores ou categorias dos parâmetros da SA 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 foram realizadas através de estatísticas descritivas. A melhor configuração de parâmetros propiciou à SA eficácia média de 95,36%, valor mínimo de 83,66%, valor máximo de 100% e coeficiente de variação igual a 3,18% do ótimo matemático obtido pelo algoritmo exato *branch and bound*. Para o problema de maior porte, a eficiência da SA foi dez vezes superior à eficiência do algoritmo exato *branch and bound*. O bom desempenho desta heurística reforçou as conclusões, tiradas em outros trabalhos, do seu enorme potencial para resolver importantes problemas de gerenciamento florestal de difícil solução pelos instrumentos computacionais da atualidade.

Palavras-chave: Gerenciamento florestal, metaheurísticas e *Simulated Annealing*.

SIMULATED ANNEALING METAHEURISTIC TO SOLVE FOREST PLANNING PROBLEM WITH INTEGER CONSTRAINTS

ABSTRACT - *The objectives of this work was to develop and test an algorithm based on Simulated Annealing (SA) metaheuristic to solve problems of forest management with integer constraints. The algorithm SA developed was tested in five problems containing between 93 and 423 decision variables, periodically subject to singularity constraints, minimum and maximum production. The problems had the objective of maximizing the net present value. SA was codified into delphi 5.0 language and the tests were performed in a microcomputer AMD K6II 500 MHZ, with RAM memory of 64 MB and hard disk of 15GB. The SA performance was evaluated according to the efficacy and efficiency measures. The different values or categories for the SA parameters were tested and compared in relation to their effects on the algorithm efficacy. The selection of the parameters' best configuration was performed by using the L&O test at 1% probability and analyses via descriptive statistics. The parameters' best configuration provided for SA average efficacy of 95.36%, minimum value equal to 83.66%, maximum value equal to 100%, with coefficient of variation of 3.18% of the mathematical optimum, obtained by the exact algorithm branch and bound. As for the larger problem, the efficiency of SA was ten times superior to the efficiency of the exact algorithm branch and bound. SA came out as a quite attractive new approach in forest management for solving important problems of difficult solution through the use of the current computational instruments.*

Key words: *Forest management, metaheuristics and Simulated Annealing.*

¹ Recebido para publicação em 16.5.2002 e aceito para publicação em 30.4.2004.

² Prof. Adjunto I do Departamento de Engenharia Rural da Universidade Federal do Espírito Santo – UFES, <flavio@cca.ufes.br>; ³ Prof. do Dep. de Engenharia Florestal da Universidade Federal de Viçosa – UFV, 36570-000 Viçosa-MG;

⁴ Prof. do Dep. de Informática da UFV.

1. INTRODUÇÃO

A metaheurística *Simulated Annealing* (SA) é um algoritmo derivado a partir de processos de recozimento sólidos desenvolvido por Metropolis, em 1953 (Youssef, 2001). A analogia com a otimização combinatória foi introduzida por Kirkpatrick et al. (1983) e aperfeiçoada por Cerny (1985). O termo *annealing* refere-se a um processo de resfriamento térmico que começa pela liquidação de um cristal, à alta temperatura, seguido pela lenta e gradativa diminuição de sua temperatura, até que o ponto de solidificação seja alcançado, quando o sistema atinge um estado de energia mínima (Ignacio et al., 2000). A SA pode ser utilizada na solução de diversos problemas de otimização combinatória. Muitos problemas de otimização florestal são de natureza combinatorial e, portanto, candidatos naturais à solução por esta técnica. Alguns exemplos incluem problemas com restrições de singularidade, adjacência, roteamento de veículos no transporte florestal, corte na indústria de móveis e de papel, dentre outros. No entanto, existem poucos trabalhos a respeito da utilização dessa técnica na solução de tais problemas. Lockwood & Moore (1992) desenvolveram um algoritmo SA para resolver um problema de planejamento de colheita florestal com restrições de adjacência no nordeste do Canadá. Em vez de gerar apenas soluções viáveis, o algoritmo desenvolvido usou uma série de funções de penalidades que diminuam o valor da função-objetivo para penalizar soluções inviáveis, desencorajando a escolha dessas soluções. Boston & Bettinger (1999) utilizaram a SA para resolver problemas de planejamento florestal com restrição de singularidade e adjacência e compararam-na com duas outras técnicas heurísticas: programação inteira monte carlo e busca tabu. Foram geradas 500 soluções para cada um dos quatro problemas, que possuíam entre 3.000 e 5.000 variáveis inteiras 0-1. A SA foi a técnica que encontrou o maior valor (cerca de

99%) para a função-objetivo, para três dos quatro problemas estudados. No Brasil, o primeiro trabalho foi o de Rodrigues (2001), que desenvolveu e testou as metaheurísticas algoritmo genético, busca tabu e SA em quatro problemas de planejamento florestal com restrições de singularidade, produção mínima e produção máxima.

Apesar de os estudos sobre o uso da metaheurística SA para resolver problemas de planejamento florestal terem indicado grandes desempenhos dessa técnica na solução de problemas de planejamento florestal, existem ainda alguns aspectos que necessitam ser melhorados para a consolidação dessa técnica na solução desses problemas. Tais melhorias podem incluir, por exemplo, novas estratégias de movimentos para melhorar o desempenho do método em problemas maiores, estudos adicionais sobre o efeito dos seus parâmetros no desempenho da heurística, critérios de parada mais informados, dentre outros. Assim sendo, idealizou-se este trabalho com os seguintes objetivos:

- Desenvolver e testar um algoritmo com base na metaheurística *Simulated Annealing* para resolver problemas de planejamento florestal.
- Expandir a aplicação do algoritmo para problemas maiores e mais complexos, através do desenvolvimento de um código computacional para esta finalidade.

2. MATERIALE MÉTODOS

2.1. O Problema

O algoritmo SA desenvolvido neste trabalho foi testado em quatro problemas de planejamento florestal (Quadro 1). Os problemas diferenciam-se pela diversificação no número de variáveis de decisão e no número e na magnitude das restrições de produção (demanda) mínimas e máximas.

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

Table 1 – Overall description of the problems selected for SA applications, through the forest management system SysFlor

Problema	Nº de Unidades de Manejo	Nº Variáveis de Decisão	Horizonte de Planejamento*	Produção Mínima (m ³)	Produção Máxima (m ³)
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.

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

2.2. O Modelo

Os problemas foram formulados com 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:

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

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

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

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

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 o modelo de programação inteira, a maximização do lucro (1) está sujeita às restrições de

singularidade (2) e (5) e à 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 *Simulated Annealing*

O algoritmo SA foi implementado no programa *SMAP-F* (Soluções Metaheurísticas Aplicadas ao Planejamento Florestal), codificado no 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, pode ser resumido conforme o fluxograma da Figura 1. Os detalhes desse algoritmo estão descritos nos tópicos seguintes.

A entrada de dados do algoritmo SA é um arquivo no formato texto gerado pelo sistema de suporte à decisão SysFlor (Rodrigues et al., 2000), contendo o modelo de programação inteira representativo do problema em foco. Além do modelo para representar o problema, a solução de um problema utilizando SA requer a definição dos valores dos seus diferentes parâmetros.

2.2.1. Codificação da Solução

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

Uma solução (x), viável segundo a restrição de singularidade, para o problema formulado através do modelo de programação inteira com as variáveis de decisão na forma binária ($X_{ij} \in \{0, 1\}$), 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 SA é gerada aleatoriamente. Com uma probabilidade uniformemente distribuída, uma variável de decisão X_{ij} é selecionada aleatoriamente em cada unidade de manejo para receber

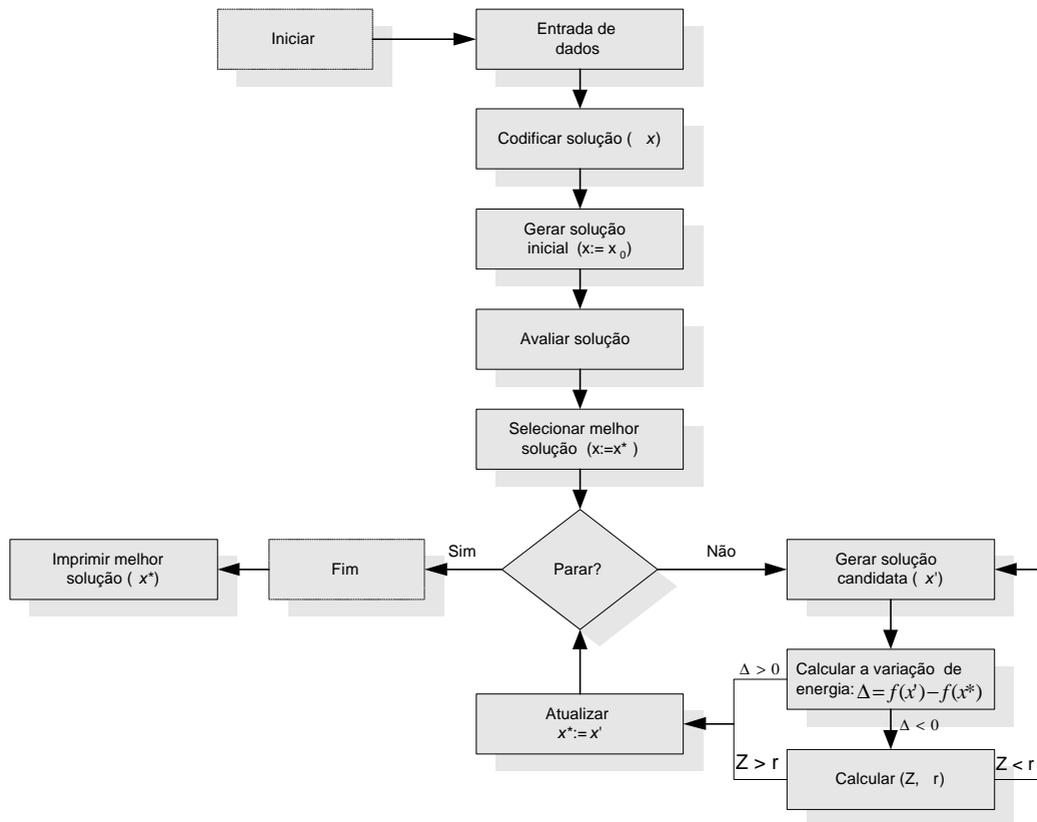


Figura 1 – SMAP-F: estrutura geral do programa para implementação da SA.
Figure 1 – SMAP-F: overall program structure with respective algorithm of SA.



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 of a solution for the problem with the decision variables in binary form.

valor igual a 1 (variável básica). As demais variáveis são assinaladas com valores iguais a zero (variáveis não-básicas), garantindo a viabilidade da restrição de singularidade, ou seja, adotar uma única opção de manejo em cada talhão.

2.2.4. Função de Avaliação

A heurística SA utiliza o valor da função-objetivo penalizada ($f_p(x)$) como função de avaliação para guiar o processo de busca de novas soluções. Esta função permite que soluções inviáveis sejam aceitas, porém aplicando-se a estas penalidades proporcionais às violações provocadas nas restrições de produção mínima e máxima. A função-objetivo $f_p(x)$ é obtida através da modificação da função-objetivo ($f(x)$), da seguinte forma:

$$f_p(x) = f(x) - vp \times VT \tag{6}$$

em que $f_p(x)$ = valor da função-objetivo penalizada para a solução x ; $f(x)$ = valor da função-objetivo do problema; vp = penalização ($\$/m^3$) para cada unidade de produção violada; VT = violação total (m^3) das restrições de produção (mínima e máxima), conforme definido por Rodrigues (2001); e demais variáveis; conforme já definidas.

2.2.5. Movimentos

Os movimentos são estratégias utilizadas para geração de soluções candidatas. Na metaheurística SA apenas uma solução candidata (x') vizinha a x ($x' \in N(x)$) é avaliada em cada iteração. Diversas estratégias de movimento podem ser definidas, e muitas vezes o seu desenho depende da natureza do problema e da criatividade do projetista. A estratégia utilizada neste trabalho usa um mecanismo aleatório para geração desse movimento. O mecanismo de movimento adotado utiliza a seguinte estratégia: a partir de uma solução inicial (x_0) gerada aleatoriamente, a cada iteração, um determinado número de unidades de manejo é escolhido aleatoriamente para ter uma variável X_{ij} adicionada (*Adc*) ($X_{ij} = 1$). Ao parâmetro número de unidades de manejo escolhido para se fazer a troca de variáveis denominou-se (*Ut*). A variável escolhida para ser removida da base (*Rem*) ($X_{ij} = 0$) é aquela cujo valor na solução anterior que teve X_{ij} adicionada é igual a 1, satisfazendo, desta forma, a restrição de singularidade. No Quadro 2 está o resumo deste procedimento com um movimento x' gerado através da adição da variável X_{11} e remoção da variável X_{12} , que era uma variável básica na solução anterior (x) na unidade de manejo 1. Neste caso, apenas uma unidade de manejo é escolhida para fazer a troca.

A heurística SA aceita a nova solução x' se um dos seguintes critérios for satisfeito:

- se o valor da função-objetivo $fp(x') > fp^*(x)$; e
- se o valor da constante Z for maior que um número aleatório (r) distribuído uniforme no intervalo entre 0 e 1, sendo:

$$Z = e^{\{fp(x') - fp^*(x)\}/T} \tag{7}$$

em que $fp(x')$ = o valor da função-objetivo penalizada da

solução corrente; $fp^*(x)$ = valor da melhor função-objetivo penalizada encontrada até o momento; T = temperatura; e = constante (2,718282); e $r \in$ uniforme{0,1}.

Uma vez que o movimento x' tenha sido aceito, a solução x na iteração seguinte passa a ser $x := x'$. O algoritmo SA guarda também a melhor solução (x^*) encontrada até o momento, sendo a sua imagem $fp^*(x)$ utilizada no cálculo de Z .

A unidade de manejo que recentemente tenha sido escolhida para ter uma troca efetuada pode permanecer por um dado período sem que nenhuma nova troca seja efetuada nesta unidade. Este é definido como período sem troca (*Pst*) e refere-se ao número de iterações na qual uma dada unidade de manejo permanecerá sem que nenhuma troca seja efetuada em suas variáveis, o que equivale a proibir que a variável recentemente adicionada (*Adc*) seja removida (*Rem*) da base.

2.2.6. Temperatura

A temperatura (T) é um dos principais parâmetros da SA. Ela começa com um valor alto, permitindo que movimentos de pior qualidade (ou que piorem a função-objetivo) tenham maiores chances de ser escolhidos, sendo gradativamente reduzida até a convergência do algoritmo, o que equivale a diminuir as chances de escolha de um movimento de baixa qualidade, à medida que o algoritmo progride. Várias estratégias de redução da temperatura podem ser adotadas, e nenhum trabalho até o momento apresentou sugestões referentes ao melhor esquema de redução dessa temperatura. Há um consenso na literatura especializada de que este parâmetro depende da natureza do problema, devendo ser definido por experimentação.

Quadro 2 – Geração de movimentos através do mecanismo de adicionar e remover aleatoriamente pares de variáveis de decisão (X_{ij}) em cada unidade de manejo

Table 2 – Generation of movements through the mechanism of randomly adding and removing decision variable pairs (X_{ij}) in each management unit

UM (i)	1			2			3			4			Movimento	
VD	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{31}	X_{32}	X_{33}	X_{41}	X_{42}	X_{43}	<i>Adc</i>	<i>Rem</i>
x	0	1	0	1	0	0	0	0	1	1	0	0	-	-
x'	1	0	0	1	0	0	0	0	1	1	0	0	X_{11}	X_{12}



Três parâmetros são utilizados para definir a temperatura (T) em um dado instante: a temperatura inicial (T_0), a taxa de redução da temperatura (R_T) e o número de iterações na qual a temperatura permanece constante (K_{TC}). Um possível esquema poderia fornecer a temperatura em um determinado instante como função potencial de T_0 , T_R e do número de iteração k , da seguinte forma:

$$T_k = T_0 \left(1 - \frac{T_R}{100} \right)^k \quad (8)$$

em que T_k = temperatura na iteração k ; T_0 = temperatura inicial; T_R = taxa de redução da temperatura, expressa em porcentagem; e k = iteração.

2.2.7. Critério de Parada

O algoritmo SA proposto utiliza o seguinte critério de parada: após encontrar a primeira solução com violação total nula ($VT=0$), a heurística SA executa um número de iterações adicionais (Δk) para tentar melhorar a solução encontrada. Assim, a SA tenta encontrar várias soluções viáveis, escolhendo aquela de melhor valor para a função-objetivo ($fp^*(x)$). Para interromper a corrida da SA após um número de iterações, sem encontrar uma solução viável para o problema, foi definido um número máximo de iterações (k_{max}).

2.3. Escolha dos Parâmetros da SA

Os parâmetros da metaheurística SA avaliados neste trabalho foram a temperatura inicial (T_0), a taxa de redução da temperatura (T_R) e o número de iterações com a

temperatura mantida constante (K_{TC}). Além desses parâmetros básicos, a metaheurística SA desenvolvida utiliza outros parâmetros com diferentes finalidades. Na estratégia de movimentos dessa metaheurística foram utilizados os parâmetros número de unidades de manejo alteradas (Ut) e período sem troca (Pst), com as respectivas finalidades de definir o número de unidades de manejo escolhidas para se fazer a troca de variáveis e o número de iterações para a qual a unidade de manejo recentemente trocada permaneça sem que nenhuma troca seja efetuada. Outros parâmetros da SA foram o valor da penalização (vp), utilizado nos cálculos das penalidades; o número de iterações adicionais (Δk) para melhorar a qualidade da solução e o número máximo de iterações permitidas (k_{max}), utilizados pelo critério de parada.

Os valores dos parâmetros utilizados neste trabalho estão apresentados no Quadro 3. As combinações entre os níveis dos diferentes parâmetros gerou um ensaio com 54 diferentes combinações, representando cada uma delas um cenário ou tratamento diferente. Cada tratamento foi repetido cinco vezes, resultando em 270 observações para cada problema. Considerando os quatro problemas utilizados no teste, gerou-se um ensaio com 1.080 observações.

2.4. Avaliação do Desempenho da SA

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

Quadro 3 – Resumo dos parâmetros utilizados pela metaheurística *Simulated Annealing*
Table 3 – Summary of the parameters used by *Simulated Annealing metaheuristics*

Parâmetro	Nível	Valor do Parâmetro		
Iterações com temperatura constante (K_{TC})	3	1	50	100
Temperatura inicial (T_0)	3	10^5	10^6	10^7
Taxa de redução da temperatura (T_R) - %	3	0,5	1,0	5,0
Unidades de manejo trocadas (Ut)	1	1		
Período sem troca (Pst)	2	0	M/2	
Valor da penalização (vp) – \$/m ³	1	10		
Número máximo de iterações (k_{max})	1	3.000		
Iterações adicionais (Δk)	1	500		

em que M = número de unidades de manejo do problema.

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

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

A eficiência da SA foi avaliada, utilizando-se apenas medidas de tempo de processamento do algoritmo, o que permite somente comparações relativas entre os métodos de solução, processadas ao utilizar os mesmos recursos computacionais. Os efeitos dos diferentes parâmetros foram avaliados pelo teste L&O, proposto por Leite & Oliveira (2002).

3. RESULTADOS E DISCUSSÃO

O comportamento da temperatura ao longo de uma corrida do algoritmo SA, considerando taxas de redução da temperatura de 0,5, 1,0 e 5,0%, está na Figura 4. Neste exemplo, o valor da temperatura inicial foi igual a 1.000.000, sendo reduzida a cada iteração, ou seja, o valor do parâmetro número de iterações com temperatura constante é igual a 1 ($K_{TC}=1$).

A temperatura é o principal parâmetro da AS, podendo afetar o seu desempenho. Temperaturas altas aumentam a probabilidade de um movimento de pior qualidade (valor inferior para função objetivo) ser aceito, o que confere maior diversidade às soluções no início da corrida. A diminuição na temperatura tem um efeito contrário, limitando as chances de os movimentos de pior qualidade serem selecionados, o que reduz a diversidade das soluções candidatas, forçando o algoritmo a convergir para um ótimo local ou global. Valores muito altos de temperatura são importantes para manter a diversidade do espaço de solução, mas podem dificultar o algoritmo a convergir para uma solução boa. Valores muito baixos, ao contrário, forçam a convergência rápida do algoritmo, mas podem levá-lo a estacionar em ótimos locais. Portanto, o equilíbrio desses fatores pode ser decisivo para o sucesso do algoritmo. Não existem na literatura especializadas recomendações seguras a respeito do melhor esquema, tanto para temperaturas iniciais quanto para o programa ideal de redução. Tais escolhas são ainda baseadas na experimentação. Os valores ideais são aqueles que garantem um equilíbrio entre a diversificação e a intensificação da busca. Uma contribuição recente em relação à adoção de procedimentos mais sólidos para

escolha da temperatura inicial pode ser encontrada em Youssef et al. (2001).

Não houve diferenças estatísticas significativas das temperaturas iniciais adotadas na eficácia da SA pelo teste de L&O, a 1% de probabilidade. Na Figura 5 está a medida de eficácia do algoritmo SA para os três diferentes valores de temperaturas iniciais adotados.

Em relação às taxas de redução da temperatura empregada, à exceção das taxas de redução iguais a 0,5 e

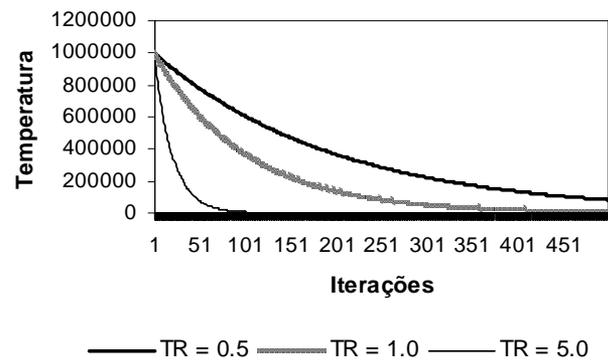


Figura 4 - Variação da temperatura em uma corrida da SA, considerando uma temperatura inicial igual a 1.000.000, taxas de redução iguais a 0,5, 1,0 e 5,0% e intervalo com a temperatura mantida constante igual a um ($K_{TC} = 1$).

Figure 4 - Temperature variation in a SA running, considering an initial temperature equal to 1,000,000, reduction rates equal to 0,5, 1,0 and 5,0% and interval with the temperature kept constant equal to one ($K_{TC} = 1$).

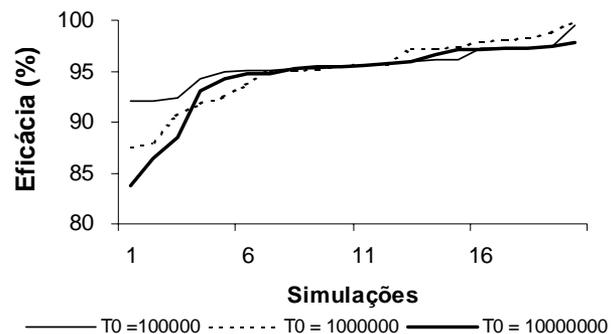


Figura 5 - Eficácia da SA para os quatro problemas avaliados, considerando diferentes valores de temperatura inicial.

Figure 5 - Efficacy of SA for the four problems analyzed, considering different initial temperature values.

1,0%, as demais diferiram entre si pelo teste de L&O, a 1% de probabilidade. A melhor taxa de redução da temperatura, para os quatro problemas avaliados, foi igual a 5% (Figura 6).

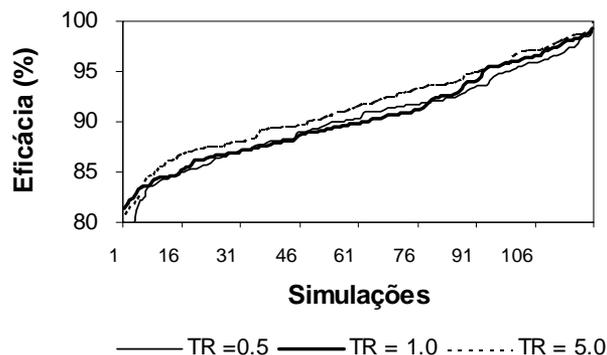


Figura 6 – Eficácia da SA para os quatro problemas avaliados, considerando diferentes valores de taxas de redução da temperatura.

Figure 6 – Efficacy of SA for the four problems analysed, considering different temperature reduction rate values.

Um outro parâmetro da SA avaliado neste trabalho foi o número de iterações com temperatura mantida constante (K_{TC}). Apenas os valores iguais a 50 e 100 não diferiram entre si pelo teste de L&O, a 1% de probabilidade. O valor do número de iterações com temperatura mantida constante igual a 1 ($K_{TC} = 1$) resultou em melhor eficácia para a SA (Figura 7). Estes resultados mostram um aspecto interessante da heurística SA, uma vez que o simples fato de mudar a estratégia de redução da temperatura ao longo da busca pode afetar o desempenho do algoritmo. Boston & Bettinger (1999), avaliando o efeito desses parâmetros para a heurística SA aplicada a problemas de planejamento florestal com restrições de singularidade, produção mínima e adjacência, com tamanho entre 3.000 e 5.000 variáveis de decisão, também encontraram resultados diferenciados. Avaliando o efeito de temperaturas iniciais iguais a 500.000, 900.000 e 1.500.000, cada uma delas mantida constante durante três períodos distintos de 100, 200 e 300 iterações, com taxas de redução após cada um desses períodos de 1, 2,5, e 5%. Os autores concluíram que temperaturas mais altas (900.000 e 1.500.000) propiciaram melhor desempenho para a heurística e não apresentaram diferenças significativas entre si. Os melhores intervalos para temperaturas mantidas constantes foram 200 e 300 iterações, sem

diferenças entre si. A melhor taxa de redução foi de 1%. Infelizmente, não há uma regra simples para determinar a melhor combinação dos diferentes parâmetros da metaheurística SA para todos os problemas. A melhor configuração para um dado problema é ainda dependente da experimentação.

Não houve efeito estatisticamente significativo, pelo teste de L&O a 1%, do parâmetro “período sem troca” (P_{st}) na eficácia da SA. Apesar de não ter sido estatisticamente significativo, este parâmetro representou uma das inovações da metaheurística SA desenvolvida neste trabalho, em relação aos principais modelos reportados na literatura. Da mesma forma que o parâmetro permanência, utilizado pela busca tabu, o objetivo principal desse parâmetro é proibir a execução de movimentos de adição e remoção de variáveis em uma unidade de manejo que tenha sido recentemente alterada (isto é, caso tenha ocorrido um movimento de trocas de variáveis através do movimento de adição e remoção na respectiva unidade de manejo). O fato de proibir movimentos em algumas unidades de manejo direciona a busca para novas regiões do espaço de solução e funciona como um mecanismo de diversificação da busca.

Os valores mínimos, médios e máximos de eficácia da SA, considerando todas as combinações de valores de temperatura inicial (T_0), taxas de redução da temperatura (R_T) e número de iterações com temperaturas mantidas constantes (KTC), foram iguais a 79,54, 90,77 e 100%, respectivamente. Nestes casos, a eficácia da SA apresentou um coeficiente de variação igual a 4,93%. Entretanto, selecionando-se apenas a melhor taxa de redução da temperatura ($TR = 5\%$) e o melhor número de iterações

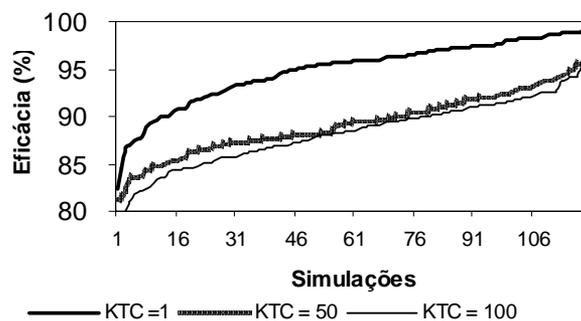


Figura 7 – Eficácia da SA para os quatro problemas avaliados, considerando diferentes valores de iterações mantidas constantes.

Figure 7 – Efficacy of SA for the four problems analyzed, considering different interaction values kept constant.

com temperatura mantida constante ($KTC = 1$), a SA apresentou eficácia mínima, média e máxima de 83,66, 95,36 e 100%, respectivamente (Figura 8). O coeficiente de variação para esta configuração de parâmetros foi igual a 3,18%. Portanto, houve melhoria no desempenho do SA, quando foi feita uma melhor escolha de seus parâmetros.

A eficiência da SA não foi avaliada rigorosamente neste trabalho, o que iria requerer análises de algoritmos mais sofisticadas, como complexidade de tempo e de espaço. Entretanto, os resultados computacionais preliminares já mostraram as vantagens da SA em relação ao algoritmo exato *branch and bound*. Na Figura 9 está apresentado o tempo de processamento da SA e do algoritmo *branch and bound*. Pode-se observar que as

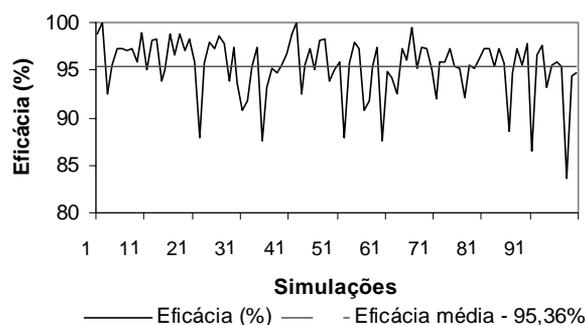


Figura 8 – Eficácia da SA nos quatro problemas analisados, considerando a melhor taxa de redução da temperatura ($R_T = 5\%$) e o melhor número de iterações com temperatura mantida constante ($KTC = 1$).

Figure 8 – Efficacy of SA in the four problems analyzed, considering the best temperature reduction rate ($R_T = 5\%$) and best number of interactions with temperature kept constant ($KTC = 1$).

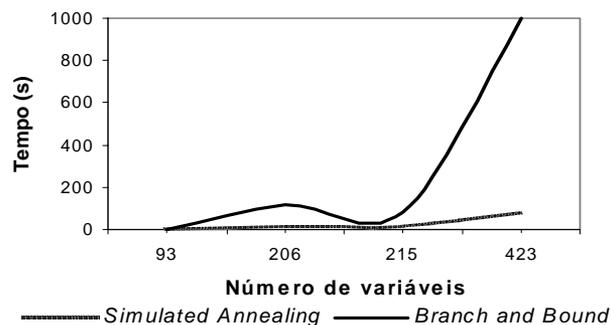


Figura 9 – Comparações do tempo de solução da metaheurística *Simulated Annealing* e o algoritmo *branch and bound*.

Figure 9 – Comparisons of the solution time for the *Simulated Annealing* metaheuristics and *branch and bound* algorithm.

diferenças entre o tempo de processamento dos dois algoritmos aumenta à medida que cresce o número de variáveis de problema. O algoritmo *branch and bound* gastou cerca de dez vezes mais tempo do que a SA para encontrar a solução do problema com 423 variáveis. Cabe ressaltar que esta vantagem pode ser melhorada através de aperfeiçoamento na estrutura do algoritmo desenvolvido, o que não foi explorado neste trabalho.

4. CONCLUSÕES

A eficácia da SA foi estatisticamente influenciada pelos valores ou categorias de alguns parâmetros empregados, avaliada pelo teste L&O, a 1% de probabilidade.

As temperaturas iniciais avaliadas não influenciaram significativamente a eficácia da SA. A melhor taxa de redução da temperatura foi igual a 5%, comparada aos valores iguais a 0,5 e 1,0%. A estratégia de redução da temperatura forneceu melhores valores de eficácia quando o número de iterações mantidas constantes foi igual a 1, comparado aos valores 50 e 100.

Utilizando os melhores parâmetros da SA, foram obtidas soluções de melhor qualidade (cerca de 95% do valor ótimo matemático).

O tempo de processamento do algoritmo SA foi bastante inferior (neste estudo, cerca de dez vezes) ao principal algoritmo exato de solução (*branch and bound*).

Com base nos resultados deste trabalho, pôde-se concluir que a metaheurística SA é flexível por comportar restrições de diversas naturezas, sem a necessidade de alterar a estrutura do algoritmo adotado.

A aplicação da SA em problemas reais é dependente de melhorias na capacidade do algoritmo em resolver problemas de grande porte, típicos do setor florestal.

O melhor desempenho da SA depende da escolha de uma boa configuração de seus parâmetros. A inexistência de conclusões seguras para a maioria dos problemas de planejamento florestal, em especial os problemas maiores e, ou, mais complexos, indica que essa escolha deve ser baseada na experimentação.

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 concedido.

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 Applications**, v. 45, n. 1, p. 41-51, 1985.
- IGNACIO, A. A. V.; FERREIRA FILHO, V. J. M.; GALVÃO, R. D. Métodos heurísticos num entorno paralelo. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 32., 2000, Viçosa. **Anais...** Viçosa: Universidade Federal de Viçosa, 2000. p. 769-788.
- JOHNSON, K. N.; SCHEURMAN, H. L. Techniques for prescribing optimal timber harvest and investment under different objectives - discussion and synthesis. **Forest Science**, v. 18, n. 1, p. 1-31, 1977.
- KIRKPATRICK Jr., S.; GELATT, C.; VECCHI, M. Optimization by simulated annealing. **Decision Science**, v. 220, n. 4598, p. 498-516, 1983.
- 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, v. 33, n. 7&8, 2002. p. 1105-1118.
- LOCKWOOD, C.; MOORE, T. Harvest scheduling with spatial constraints: a simulated annealing approach. **Canadian Journal Forest Research**, v. 23, p. 468-478, 1992.
- RODRIGUES, F. L.; LEITE, H. G.; ALVES, J. M. *Sysflor* – um sistema de suporte à decisão em planejamento florestal. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 32., 2000, Viçosa. **Anais...** Viçosa: Universidade Federal de Viçosa, 2000. p. 974-995.
- 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.
- YOUSSEF, H.; SAIT, S. M.; ADICHE, H. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. **Engineering Applications of Artificial Intelligence**, v. 14, p. 167-181, 2001.