

METAHEURÍSTICA ALGORITMO GENÉTICO 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 Gilson Fernandes da Silva²

RESUMO - Os objetivos deste trabalho foram desenvolver e testar um algoritmo genético (AG) para a solução de problemas de gerenciamento florestal com restrições de integridade. O AG 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 AG foi codificado em ambiente *delphi* 5.0 e os testes foram realizados em um microcomputador AMD K6II 500 MHZ, com memória RAM de 64 MB e disco rígido de 15GB. O desempenho do AG foi avaliado de acordo com as medidas de eficácia e eficiência. Os valores ou categorias dos parâmetros do AG 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 ao AG eficácia média de 94,28%, valor mínimo de 90,01%, valor máximo de 98,48%, com coeficiente de variação de 2,08% do ótimo matemático, obtido pelo algoritmo exato *branch and bound*. Para o problema de maior porte, a eficiência do AG foi cinco vezes superior à eficiência do algoritmo exato *branch and bound*. O AG apresentou-se como uma abordagem bastante atrativa para solução de importantes problemas de gerenciamento florestal.

Palavras-chave: Gerenciamento florestal, metaheurísticas e algoritmos genéticos.

GENETIC ALGORITHM METAHEURISTIC TO SOLVE FOREST PLANNING PROBLEM WITH INTEGER CONSTRAINTS

ABSTRACT - The objectives of this work was to develop and test a Genetic Algorithm (GA) to solve problems of forest management with integer constraints. GA was tested in five problems containing 93 - 423 decision variables, periodically subject to singularity constraints, minimum and maximum production. The problems had the objective of maximizing the net present value. GA 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 GA performance was evaluated according to the efficacy and efficiency measures. The different values or categories for the GA 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 GA average efficacy was of 94.28%, minimum value equal to 90.01%, maximum value equal to 98.48%, with coefficient of variation of 2.08% of the mathematical optimum, obtained by the exact algorithm **branch and bound**. As for the larger problem, the efficiency of GA was five times superior to the efficiency of the exact algorithm **branch and bound**. GA was found to be a quite attractive approach to solve important forest management problems.

Key words: Forest management, metaheuristics and genetic algorithm.

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

As heurísticas têm se destacado como as abordagens mais promissoras para solução de problemas de programação inteira. Enquanto os algoritmos exatos garantem uma solução ótima para certos tipos desses problemas, os métodos heurísticos não podem provar a otimalidade das suas soluções, mas oferecem soluções aceitáveis, inclusive para problemas complexos e de grande porte, com baixo custo computacional (Ignízio & Cavalier, 1994). Apesar das inúmeras aplicações dessas técnicas em diversas áreas da atividade humana, elas têm sido pouco utilizadas no meio florestal. Alguns exemplos incluem os trabalhos de Yoshimoto & Brodie (1994a), Weintraub et al. (1994), Murray & Church (1996) e Nobre (1999). Por outro lado, técnicas heurísticas mais sofisticadas, como as chamadas metaheurísticas, com aplicações variadas em diferentes áreas da ciência, têm sido ainda pouco exploradas. 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 a grande eficiência dessas técnicas são as suas principais vantagens em relação às heurísticas convencionais. Existem poucos trabalhos a respeito da aplicação das metaheurísticas em problemas de planejamento florestal, podendo-se citar dentre eles Lockwood & More (1993), Laroze & Gleber (1997) Boston & Bettinger (1999) e Laroze (1999). No Brasil, estudos relatando a utilização de metaheurísticas para solução de problemas de gerenciamento podem ser encontrados em Rodrigues (2001). A linha de pesquisa deste autor envolve a utilização das metaheurísticas AG, BT e SA para solução de problemas de planejamento florestal com restrições de integridade.

Dentre as metaheurísticas, os algoritmos genéticos (AGs), inspirados por mecanismos de evolução natural e genética, têm se destacado como técnica de otimização robusta e eficiente (Holland, 1975; Goldberg, 1989). Os AGs têm sido utilizados na solução de problemas de diferentes domínios, como programação automática, ecologia, pesquisa operacional etc. (Garcia et al., 2000). Para Rebello & Hamacher (2000), os AGs são métodos robustos que podem ser utilizados para resolver problemas em pesquisa numérica, otimização de funções e aprendizagem de máquinas, entre outros. Eles têm sido

apontados como técnicas promissoras para solucionar problemas combinatoriais, como os clássicos problemas de programação inteira. Apesar de não garantirem otimalidade, possuem como principal vantagem melhor desempenho computacional, em relação aos métodos concorrentes, além de ser um procedimento de relativa simplicidade (GARCIA et al., 2000). Segundo estes autores, praticamente todos os problemas NP-completos ou NP-difíceis têm uma versão de solução heurística que usa esse tipo de algoritmo.

Muitos problemas de planejamento florestal apresentam características próprias para solução mais eficiente por meio dos AGs. Exemplos de problemas combinatoriais em planejamento florestal são: regulação florestal com escolha de um único regime por talhão, problemas de adjacência em talhões selecionados para colheita, seleção de rotas de transporte de madeira, seccionamento de toras, corte em indústria moveleira, corte de papel, dentre outros. Neste trabalho, foi desenvolvido e testado um AG com o propósito de resolver uma instância de problemas de regulação florestal. Para avaliar os efeitos de alguns dos parâmetros do AG na estratégia de solução, foi desenvolvido um código computacional.

2. MATERIAL E MÉTODOS

2.1. O Problema

Para testar o AG desenvolvido, foram utilizados quatro problemas de planejamento florestal, conforme descrito no 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. As variáveis de decisão representam as alternativas de manejo, uma seqüência de estados da floresta que ocorrem durante o horizonte de planejamento. A geração das alternativas de manejo foi efetuada, variando-se a idade de corte e o regime de manejo, permitindo ciclos de um ou dois cortes. A abordagem utilizada para a formulação das alternativas de manejo foi o modelo I, proposto por Johnson & Scheurman (1977).

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 unidade de manejo).

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 problems 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 Matemático

Os problemas foram formulados com o uso do Sistema de Planejamento Florestal *SysFlor* (Rodrigues et al., 2000). A abordagem utilizada para as alternativas de manejo (x_{ij}) foi o modelo I, proposto por Johnson & Scheurman (1977), resultando em:

Objetivo:

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

Sujeito a:

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

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

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

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

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 (m³) 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 (m³) 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 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.

2.3. O Algoritmo Genético

O AG foi implementado dentro da rotina computacional *SMAP-F* (Soluções Metaheurísticas Aplicadas ao Planejamento Florestal), desenvolvida com o uso do ambiente de programação *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 15 GB. A estrutura geral do programa está apresentada na Figura 1 e os detalhes sucintos da implementação estão descritos nos itens subsequentes. Uma descrição detalhada desse algoritmo pode ser encontrada em Rodrigues (2001).

O modelo matemático no formato texto (*txt*) é lido diretamente pelo *SMAP-F*, para posterior solução pelo AG. Além do modelo para representar o problema a ser resolvido, o AG requer como entrada de dados a definição dos valores dos seus parâmetros.

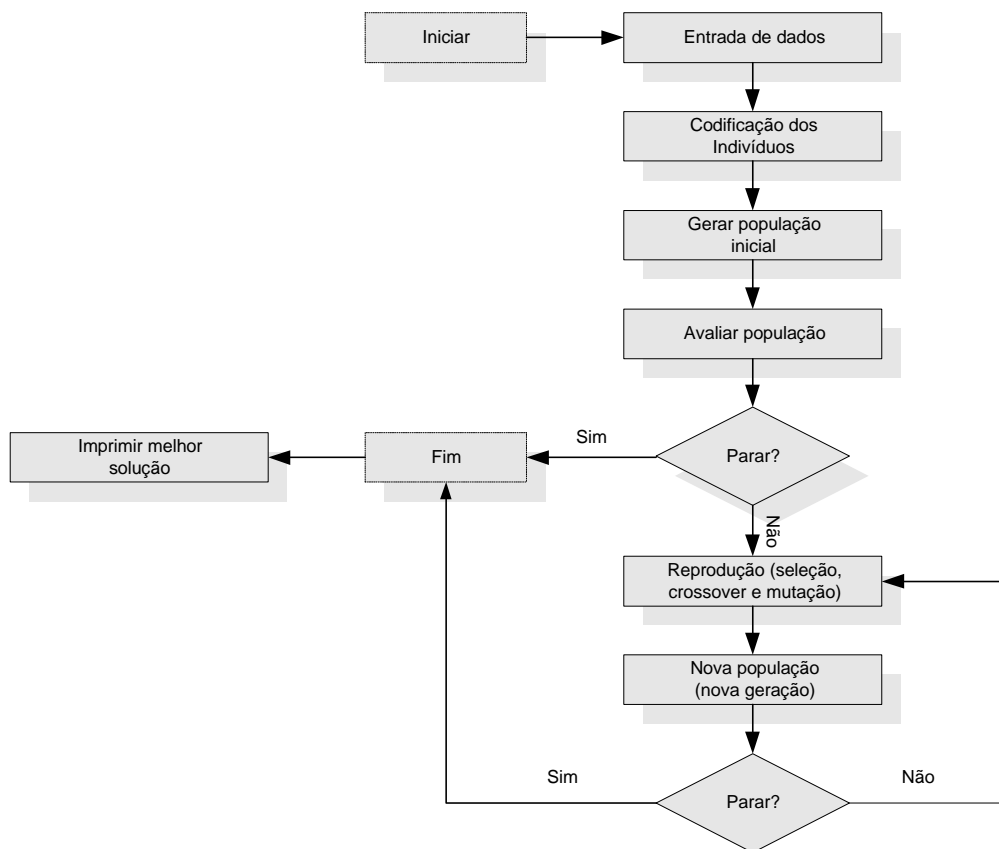


Figura 1 – SMAP-F: esquema geral do programa para implementação do AG.
Figure 1 – SMAP-F: overall program structure for implementation of AG.

2.3.1. Codificação dos Indivíduos

Antes de apresentar a codificação dos indivíduos para o AG, é importante visualizar um esquema para representar a solução para o problema em foco. Aqui, e em outros itens subsequentes, foram inseridos alguns comentários pertinentes ao tema central do trabalho, isto é, algoritmos genéticos, visando a melhor interpretação da metodologia e dos resultados apresentados no item 3. Conforme já demonstrado, para representar as alternativas de manejo do problema foi utilizado uma variável de decisão X_{ij} , simbolizando a alternativa de manejo j ($j=1,2,\dots,N$) assinalada à unidade de manejo i ($i=1,2,\dots,M$). Um vetor $V(x) = \{X_{11}, X_{12}, \dots, X_{ij}\}$ com estas respectivas variáveis de decisão pode ser representado conforme Figura 2.

Uma solução possível (x) 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 $f(x)$, é apresentada na Figura 3.

Na terminologia dos AGs cada indivíduo ou cromossomo representa uma solução para o problema. Fazendo

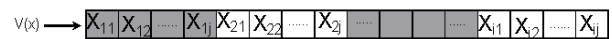


Figura 2 – Representação de um vetor com as variáveis de decisão do problema.

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

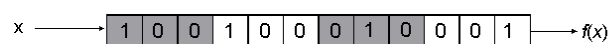


Figura 3 – Representação de uma solução para o problema com as variáveis de decisão na forma binária.

Figure 3 – Configuration of a solution for the problem with the decision variables in binary form.

uma analogia entre os indivíduos ou cromossomos e o esquema de solução para o problema, constata-se que cada *gene* equivale a uma alternativa de manejo (X_{ij}). Aos possíveis estados assumidos por cada *gene* dá-se o nome de *alelo*. À posição ou ao endereço dos *genes* no cromossomo denomina-se *locus*. Cada cromossomo é avaliado segundo uma função de aptidão ou *fitness*, usualmente igual à função-objetivo ($f(x)$) do problema (Figura 4)

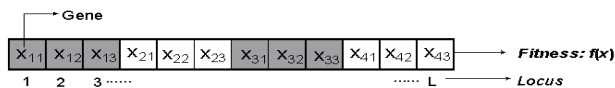


Figura 4 – Cromossomo com seus respectivos *genes*, *locus* e *fitness*.

Figure 4 – Chromosome with its respective *genes*, *locus* and *fitness*.

A codificação binária foi utilizada para representar os indivíduos, e uma particularidade na criação de cromossomo, utilizada neste trabalho, foi a de assinalar os valores dos genes de forma que os cromossomos gerados representassem apenas soluções viáveis, segundo a restrição de singularidade, ou seja, ao conjunto de genes j ($j = 1, 2, \dots, N$) que representam as variáveis de decisão em cada unidade de manejo i ($i = 1, 2, \dots, M$), um único gene recebe valor igual a 1 ($X_{ij} = 1$) e os demais recebem valor igual a zero ($X_{ij} = 0$) (Figura 5).

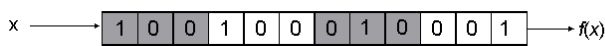


Figura 5 – Representação de um cromossomo com seus alelos na forma binária.

Figure 5 – Configuration of a chromosome with its alleles in the binary form.

2.3.2. Função de Avaliação

A função de avaliação do AG, denominada de função de aptidão ou *fitness*, é utilizada para medir o desempenho dos indivíduos, podendo ser determinada de diversas formas. A forma mais comum em problemas de otimização é a utilização da própria função-objetivo do problema. Então, nos modelos utilizados para representar os problemas de planejamento florestal, a aptidão ou *fitness* de cada indivíduo pode ser obtida por:

$$f(x) = \sum_{i=1}^M \sum_{j=1}^N c_{ij} \cdot x_{ij} \quad (6)$$

em que $f(x)$ = função-objetivo utilizada para representar a aptidão ou *fitness* dos indivíduos; x_{ij} = variável de decisão (gene) que representa a j -ésima alternativa de manejo adotada na i -ésima unidade de manejo; e c_{ij} = valor presente líquido total (\$) de cada unidade de manejo i , colhida segundo a alternativa j .

A expressão (6) é útil para avaliar aqueles indivíduos que representam soluções viáveis para o problema. Entretanto, a geração de indivíduos com estas características pode ser um processo muito oneroso, devido a dificuldades, em muitos problemas, de identificar as suas soluções viáveis. O teste da viabilidade dos indivíduos (soluções) pode ser dispensado quando são utilizadas funções de avaliação apropriadas. Uma das alternativas é permitir que os indivíduos inviáveis permaneçam na população, porém aplicando-se a eles uma penalidade. Este artifício é muito utilizado pelos algoritmos genéticos reportados na literatura, e foi também implementado neste trabalho, conforme descrito a seguir.

Considerando que as restrições de singularidade são atendidas no processo de geração dos cromossomos, o método das penalidades consiste em relaxar as demais restrições do problema (demanda mínima e máxima), alterando o cálculo da *fitness* da seguinte forma:

$$f_p(x) = \sum_{i=1}^M \sum_{j=1}^N c_{ij} \cdot x_{ij} - vp \times VT \quad (7)$$

$$VT = \sum_{k=0}^{H-1} (D_k + S_k) = \sum_{k=0}^{H-1} D_k + \sum_{k=0}^{H-1} S_k = DT + ST \quad (8)$$

$$D_k = Dmin_k - \sum_{i=1}^M \sum_{j=1}^N V_{ijk} X_{ij}; \{k = 0, 1, \dots, H - 1\} \quad (9)$$

$$S_k = \sum_{i=1}^M \sum_{j=1}^N V_{ijk} X_{ij} - Dmax_k; \{k = 0, 1, \dots, H - 1\} \quad (10)$$

em que $f_p(x)$ = valor da *fitness* (função-objetivo) penalizada (\$); VT = violação total (m^3) das restrições de produção mínima e máxima; D_k = déficit de produção (m^3) no k -ésimo período do horizonte de planejamento, quando a restrição de produção mínima é violada; S_k = superávit de produção (m^3) no k -ésimo período do horizonte de

planejamento, quando a restrição de produção máxima é violada; DT = déficit total de produção (m^3); ST = superávit total de produção (m^3); e vp = penalização ($\$/m^3$) para cada unidade de produção violada.

2.3.3. Geração da População Inicial

Ao conjunto de indivíduos utilizado para representar um conjunto de soluções para o problema em um determinado instante denomina-se população. O ponto de partida de um AG é uma população inicial (P_0), e através de repetidas operações são geradas novas populações (P_n), também conhecidas como novas gerações, até que um critério de parada seja atendido. A população inicial do AG desenvolvido neste trabalho foi gerada aleatoriamente. O processo consistiu em escolher aleatoriamente, em cada unidade de manejo, uma variável de decisão (gene) para receber o valor 1, assinalando às demais variáveis daquela unidade o valor 0, satisfazendo com isto a restrição de singularidade. Cabe lembrar que o número de indivíduos da população (n) determina o seu tamanho, sendo este um parâmetro do AG.

2.3.4. Seleção dos Indivíduos

A seleção consiste em escolher os indivíduos da população que irão se reproduzir (cromossomos-pais). Existem vários métodos de seleção descritos na literatura. Neste trabalho, foram testados os seguintes métodos de seleção: proporcional, escalonamento, *Boltzmann*, ranqueamento e *tournament*. Por exemplo, no método de seleção de *Boltzmann*, a expectativa de sobrevivência dos indivíduos é calculada por meio da seguinte expressão:

$$E_{ik} = \frac{e^{f_p(x)_{ik}/T_k}}{\sum_{i=1}^n e^{f_p(x)_{ik}/T_k}} / n \quad (11)$$

em que $f_p(x)_{ik}$ = valor da *fitness* (função-objetivo) penalizada do i -ésimo indivíduo na k -ésima iteração; T_k = temperatura na k -ésima iteração; e n = tamanho da população.

A temperatura (T) começa com um valor (T_0) alto, sendo gradativamente reduzida até a convergência do algoritmo. Vários programas de redução da temperatura podem ser adotados, e nenhum trabalho até o momento apresentou sugestões a respeito do melhor esquema de redução dessa temperatura. Dois parâmetros são utilizados para definir a temperatura (T) em um dado instante: a

temperatura inicial (T_0) e a taxa de redução da temperatura (R_T). Neste trabalho, o esquema utilizado para fornecer a temperatura em um determinado instante foi a função potencial, dada por:

$$T_k = T_0 \left(1 - \frac{R_T}{100} \right)^k \quad (12)$$

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

Detalhes sobre os demais métodos de seleção podem ser encontrados em Rodrigues (2001). Com exceção do método de seleção *tournament*, no qual os pares de indivíduos selecionados aleatoriamente são cruzados entre si, os demais métodos de seleção propostos neste trabalho necessitam de uma operação adicional para viabilizar a seleção dos indivíduos que sofrerão cruzamento. Tendo sido calculada a expectativa de sobrevivência dos indivíduos através de um método de seleção, um mecanismo probabilístico deve ser utilizado para escolha dos melhores indivíduos, levando em consideração as suas expectativas de sobrevivência. O *roleta-russa* é um dos principais métodos utilizados neste processo. Este método consiste em associar a cada indivíduo uma fatia da roleta igual à sua probabilidade de sobrevivência (P_i), dada por:

$$P_i = \frac{E_{ik}}{\sum_{i=1}^n E_{ik}} \quad (13)$$

em que E_{ik} = expectativa de sobrevivência do indivíduo na k -ésima iteração.

De acordo com essa expressão, a probabilidade de sobrevivência do indivíduo é proporcional à sua expectativa de sobrevivência. A implementação computacional deste método foi feita da seguinte forma. Após calculada a probabilidade de sobrevivência (P_i) de cada indivíduo, atribuiu-se a cada um deles uma fatia (F_i) na roleta igual à sua probabilidade de sobrevivência P_i . Os indivíduos são assinalados seqüencialmente na roleta, conforme esquematizado no exemplo da Figura 6, onde cada indivíduo i da população tem a sua fatia (F_i) na roleta compreendida no intervalo $L_{i-1} \leq F_i < L_i$ (em que $L_0 = 0$ e $L_i = L_{i-1} + P_i$). Após definidos os intervalos que cada indivíduo ocupa na roleta, um número aleatório r ($0 \leq r \leq 1$) é gerado para simular uma rodada da roleta. O indivíduo selecionado

em cada rodada da roleta é aquele cujo número aleatório gerado pertence ao intervalo do respectivo indivíduo na roleta. O processo é repetido até que a nova população seja restabelecida.

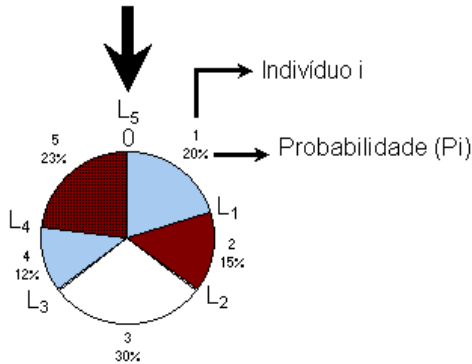


Figura 6 – Esquema de uma roleta-russa para uma população (n) de cinco indivíduos.

Figure 6 – Scheme of a roulette wheel for a population (n) with five individuals.

2.3.5. Operadores Genéticos

Os operadores genéticos (ψ) são mecanismos que garantem a evolução dos indivíduos, criando, a partir de uma dada população inicial (P_0), novas populações (P_i) ou gerações (G_i) de indivíduos melho-

rados ($P_0 \xrightarrow{y} P_1 \xrightarrow{y} P_2 \dots \xrightarrow{y} P_i \equiv G_0 \xrightarrow{y} G_1 \xrightarrow{y} G_2 \dots \xrightarrow{y} G_i$).

A cada nova população ou geração espera-se melhoria no desempenho dos indivíduos em relação ao meio ambiente, o que traduzido para o processo de busca de novas soluções pelo AG significa obter uma melhoria na qualidade das novas soluções geradas.

Os operadores genéticos utilizados neste trabalho foram o *crossover* e a mutação. O operador de *crossover* permite a obtenção de novos indivíduos (filhos), a partir da combinação (cruzamento) dos cromossomos dos pais. Utilizando um método de seleção qualquer, dois indivíduos-pais da população são selecionados e cruzados entre si para gerar dois novos indivíduos (*filhos*) para a nova geração. A probabilidade de cruzamento dos dois indivíduos-pais é conhecida como taxa de *crossover* (T_c). O cruzamento dos dois indivíduos-pais é efetuado toda vez que um número aleatório r , com distribuição uniforme, selecionado entre o intervalo 0 e 1, for menor que a taxa de *crossover* ($r \leq T_c$). Este processo é efetuado até que o

tamanho da população seja restabelecido. Os tipos de *crossover* mais utilizados nos AGs são de um único ponto, de dois pontos e de multipontos.

No *crossover* de um único ponto os dois indivíduos pais selecionados para reproduzir têm seus materiais genéticos (correspondem aos estados ou valores dos genes) trocados a partir de um ponto de ruptura, escolhido aleatoriamente (Figura 7). Para preservar a estrutura original do cromossomo, propiciando a geração de cromossomos-filhos que representem apenas soluções viáveis segundo as restrições de singularidade, foi utilizado um *crossover* especial. Neste *crossover*, restringem-se os pontos de quebra em um conjunto de pontos pertencentes ao grupo de posições que represente as interfaces das unidades de manejo (Figura 7), o que garante a geração de novos indivíduos com a características desejadas.

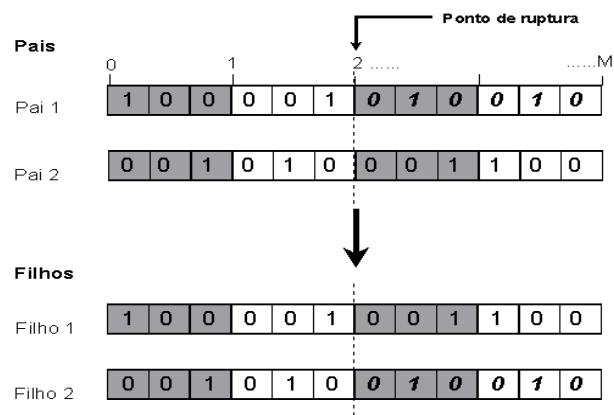


Figura 7 – Crossover de um ponto gerando novos indivíduos.

Figure 7 – Crossover of one point producing new individuals.

Após a ocorrência do *crossover*, alguns indivíduos da população podem sofrer mutação. O operador de mutação altera os valores de alguns *bits* (alelos) do cromossomo, conferindo diversidade à população. A aplicação desse operador é feita ao escolher aleatoriamente, as posições de um ou mais genes para mutação e ao inverter os valores dos seus *alelos*. A mutação é, em geral, aplicada a todos os cromossomos da população, ou seja, em 100% dos indivíduos. Para o AG desenvolvido neste trabalho foi definido um novo parâmetro para permitir a ocorrência da mutação com taxas inferiores a 100% dos indivíduos. Este parâmetro foi denominado *taxa*

de cromossomos mutados (T_{CM}) e refere-se à porcentagem dos cromossomos da população que sofrerão mutação em uma dada corrida do AG. A *probabilidade de mutação* (P_m), também conhecida como *taxa de mutação* (T_m), por outro lado, determina a porcentagem de genes mutados em cada cromossomo. A Figura 8 representa um esquema de mutação pontual, onde após escolhido o cromossomo 1 e os seus genes a serem mutados os valores de seus alelos foram invertidos de 1 para 0 e vice-versa.

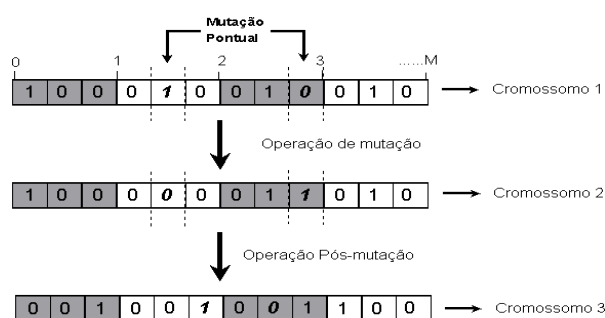


Figura 8 – Operação de mutação e pós-mutação para obtenção de cromossomos viáveis.

Figure 8 – Mutation and after - mutation operation to obtain feasible chromosomes.

O operador de mutação pode, entretanto, alterar a condição de viabilidade dos cromossomos. A Figura 8 ilustra dois problemas com a operação de mutação. No primeiro caso, após a mutação do cromossomo 1, todos os genes (isto é, as variáveis de decisão) da unidade de manejo 2 (cromossomo 2) possuem valores nulos. O segundo caso ocorre quando mais de um gene é assinalado com valor igual a 1 em uma mesma unidade de manejo, conforme pode ser visto na unidade de manejo 3 do cromossomo 2. Assim, ao desejar o assinalamento de exatamente uma alternativa de manejo (gene) em cada unidade de manejo e evitar a geração de cromossomos inviáveis, segundo a restrição de singularidade, torna-se necessário corrigir o problema. Tal correção é feita através de uma operação pós-mutação. Esta operação foi efetuada de duas formas distintas: (i) se as variáveis de decisão de uma dada unidade de manejo forem todas nulas, uma nova variável, diferente daquela selecionada para ser mutada, é escolhida aleatoriamente para receber valor 1, o que garante o assinalamento de exatamente uma variável em cada unidade de manejo; (ii) no caso de duas variáveis receberem valores iguais a 1 (violação da

restrição de singularidade), a variável que receberá valor igual a 0 (zero) é aquela que possuía valor igual a 1 antes da mutação. O cromossomo 3 da Figura 8 ilustra a operação pós-mutação para corrigir estes dois problemas.

2.3.6. Critério de Parada

Os procedimentos iterativos requerem a definição de um critério de parada para interromper a corrida do algoritmo. Segundo Youssef et al. (2001), um dos problemas enfrentados na confecção de um AG é determinar o critério de parada. Esta dificuldade se deve à inabilidade ou dificuldade desses algoritmos em avaliar a qualidade da solução em um dado instante da busca, pois, em geral, são míopes (ou seja, geralmente não há informação de quanto a solução corrente se aproxima da solução ótima). Há, contudo, alguns mecanismos para definição do critério de parada. O número máximo de iterações, o tempo máximo de processamento e a estabilização da função-objetivo são alguns exemplos. A escolha desse critério pode também afetar a qualidade da solução encontrada. Por exemplo, ao utilizar como critério de parada a primeira solução viável encontrada, uma solução de baixa qualidade pode ser escolhida, uma vez que para um dado problema pode existir uma infinidade de soluções viáveis. O AG proposto utiliza como critério de parada o seguinte procedimento: após encontrar a primeira solução viável (isto é, com violação total igual a zero, $VT = 0$), o AG executa um número de iterações adicionais (Dk) para tentar melhorar a solução encontrada. Portanto, o AG tenta encontrar várias soluções viáveis, escolhendo aquela de melhor valor para a função-objetivo. Para interromper a corrida do AG após um número de iterações, sem encontrar uma solução viável para o problema, foi definido um número máximo de iterações (k_{max}).

2.3.7. Escolha dos Parâmetros do AG

Em geral, um AG possui duas categorias de parâmetros: qualitativos e quantitativos. Os principais parâmetros qualitativos são os tipos de *crossover* (um ponto, dois pontos e múltiplos pontos) e os métodos de seleção (proporcional, escalonamento, *Boltzmann*, ranqueamento e *tournament*). Os principais parâmetros quantitativos são constituídos pelo tamanho da população (n), pela taxa de *crossover* (T_c) e pela taxa de mutação (T_m). Além dos parâmetros básicos, o AG desenvolvido neste trabalho inclui outros parâmetros utilizados com diferentes finalidades, como: o valor da penalização

(*vp*), utilizado nos cálculos das penalidades; o número de iterações adicionais (*Dk*), para melhorar a qualidade da solução utilizada no critério de parada; o número máximo de iterações permitidas (*k_{max}*); a temperatura inicial (*T₀*) e a taxa de redução de temperatura (*T_R*), utilizados pelo método de seleção de *Boltzmann*, e o valor máximo para o maior rank (*Max*), utilizado pelo método de seleção por ranqueamento.

Existem muitos trabalhos desenvolvidos com a finalidade de dimensionar os valores dos parâmetros na literatura de computação evolucionária, em muitas áreas da ciência. Entretanto, não há resultados conclusivos sobre os melhores valores desses parâmetros para todas as aplicações. Muitas aplicações usam valores reportados em outros trabalhos, outras definem os seus próprios valores com base em experimentação. Para os problemas de planejamento florestal propostos, não existem sequer trabalhos que reportam a utilização dos AGs para solucioná-los, o que indica o uso da experimentação para dimensioná-los. Os parâmetros e seus respectivos níveis ou valores testados neste trabalho estão no Quadro 2. As combinações entre os níveis e os parâmetros apresentados neste quadro geraram um ensaio com 60 diferentes combinações. Cada combinação pode ser vista como um cenário ou tratamento diferente. O objetivo é escolher o melhor cenário ou tratamento. Os valores dos parâmetros *T_C*, *T_m*, *T₀*, *T_R* e *Max* foram escolhidos com base no trabalho de Garcia et al. (2000). Os demais valores de parâmetros

foram escolhidos após um ensaio piloto, fornecendo algumas diretrizes para suas escolhas.

Para comparar os métodos de seleção e o tamanho da população, foram utilizados apenas o *crossover* de um único ponto e os demais parâmetros fixados, conforme sugerido por Garcia et al. (2000). Este ensaio foi feito para todos os problemas, utilizando quatro repetições para cada tratamento, gerando um total de 320 observações.

Os tipos de *crossover* foram comparados ao utilizar, além dos parâmetros sugeridos por Garcia et al. (2000), o melhor método de seleção e o tamanho da população previamente selecionados, neste trabalho, por experimentação. Este ensaio foi feito também para todos os problemas, utilizando oito repetições por tratamento, o que gerou um total de 96 observações.

2.3.8. Avaliação do Desempenho do AG

O desempenho do AG desenvolvido neste trabalho foi avaliado através da sua eficácia e eficiência. A eficácia foi calculada pela razão, expressa em porcentagem, entre o melhor valor da função-objetivo (*fitness*) do AG e o ótimo matemático, obtido pelo algoritmo exato *branch and bound*, assim:

$$Ef = \frac{f_{AG}}{f_o} \times 100 \tag{15}$$

Quadro 2 – Resumo dos parâmetros utilizados pelo algoritmo genético

Table 2 – Summary of the parameters used by genetic algorithm metaheuristics

Parâmetro	Nível	Valor ou Categoria do Parâmetro				
		I	II	III	IV	V
Método de seleção	5	I	II	III	IV	V
Tamanho da população (<i>n</i>)	4	25	50	75	100	
Tipo de <i>crossover</i>	3	1P	2P	MP		
Taxa de <i>crossover</i> (<i>T_C</i>) - %	1	95				
Taxa de cromossomos mutados (<i>T_{CM}</i>) - %	1	100				
Taxa de mutação (<i>T_m</i>) - %	1	1,0				
Valor da penalização (<i>vp</i>) – \$/m ³	1	10				
Temperatura inicial (<i>T₀</i>)	1	10 ⁶				
Taxa de redução de temperatura (<i>T_R</i>) - %	1	1				
Valor para rank máximo (<i>Max</i>)	1	1,1				
Número máximo de iteração (<i>k_{max}</i>)	1	500				
Iterações adicionais (<i>Δk</i>)	1	50				

em que I = seleção proporcional, II = seleção por escalonamento, III = seleção de *Boltzmann*, IV = seleção por ranqueamento, V = seleção *tournament*, 1P = *crossover* de um ponto, 2P = *crossover* de dois pontos e MP = *crossover* multipontos.



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

As medidas adequadas de eficiência dependem de estudos mais sofisticados, como a análise de complexidade de algoritmos, como complexidade de tempo e de espaço. Neste trabalho, a eficiência do AG foi avaliada utilizando-se apenas o tempo de processamento do algoritmo, o que permite apenas comparações relativas entre os métodos de solução, processadas ao utilizar os mesmos recursos computacionais.

As comparações entre os métodos de seleção, o tamanho da população e o tipo de *crossover* foram feitas com o procedimento estatístico denominado teste L&O, proposto por Leite & Oliveira (2002).

3. RESULTADOS E DISCUSSÃO

Na Figura 9 estão as medidas de eficácia dos cinco métodos de seleção avaliados. Pode-se observar que os métodos de seleção de *Boltzmann* e escalonamento apresentaram resultados superiores aos dos demais. A análise estatística desses resultados, utilizando o teste de L&O a 1%, mostrou que apenas os métodos de seleção por escalonamento e *Boltzmann* não apresentaram diferenças estatísticas significativas entre si. A eficácia foi medida pela relação entre o valor da *fitness* do AG e o ótimo matemático obtido através da solução do problema, utilizando o algoritmo exato *branch and bound*. De acordo com a Figura 9, o pior método de seleção foi o de *tournament*.

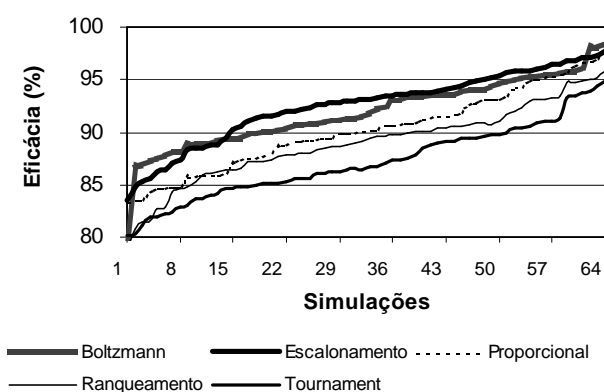


Figura 9 – Medida de eficácia do AG nos quatro problemas analisados, considerando diferentes métodos de seleção.

Figure 9 – Efficacy measurements of the AG for the four problems analyzed, considering different selection methods.

Os melhores resultados apresentados pelo método de seleção por escalonamento e *Boltzmann* já eram esperados, uma vez que eles possuem mecanismos de seleção teoricamente mais eficientes que os dos demais. Os métodos de seleção de *Boltzmann* e escalonamento utilizam mecanismos diferenciados de pressão de seleção dos indivíduos ao longo da busca, utilizando informações do problema. Enquanto a seleção por escalonamento mantém uma pressão de seleção aproximadamente constante ao longo da busca, utilizando informações do desvio-padrão da população, a seleção de *Boltzmann* altera a pressão de seleção ao longo da busca, utilizando um mecanismo de resfriamento, ou diminuição da temperatura (Mitchell, 1996). A variação da pressão de seleção pelo método de *Boltzmann* é conseguida através da diminuição da temperatura. Com a temperatura maior no início da busca, todos os indivíduos apresentam expectativa de sobrevivência igual, o que propicia uma probabilidade teoricamente igual de sobrevivência. Com a corrida do algoritmo, a pressão de seleção vai sendo aumentada através da redução da temperatura, com os indivíduos mais aptos tendo maiores chances de sobrevivência. Ao contrário da seleção por escalonamento e *Boltzmann*, a seleção *tournament* ou por jogos é totalmente aleatória, não utilizando nenhuma informação do problema para seleção dos indivíduos ao longo da busca. Os piores resultados desse método ressaltam as vantagens dos métodos mais informados.

Vale ressaltar que cuidados especiais devem ser tomados na utilização dos métodos de seleção de *Boltzmann* e ranqueamento, uma vez que tais métodos utilizam parâmetros adicionais que podem afetar o desenvolvimento do AG. Considerando que os parâmetros desses métodos foram fixados com base em experiências de outros problemas, é possível que tais escolhas não sejam as mais adequadas para um bom desempenho de o AG resolver os problemas em foco. A exemplo de outros parâmetros, estes poderiam ter sido ajustados para os problemas propostos através de experimentação prévia. O método de seleção de *Boltzmann*, além de utilizar parâmetros adicionais, pode utilizar diferentes programas de redução de temperatura, não foram avaliados neste trabalho.

Outro aspecto que merece ser destacado refere-se ao critério de parada utilizado. Recordando que o critério de parada adotado utilizou um número de iterações adicionais (Dk), para melhorar a primeira solução viável encontrada, percebe-se que a escolha deste valor pode representar um aspecto muito importante na qualidade



da solução encontrada pelo AG. Uma escolha balanceada de Δk é decisiva para o bom desempenho do AG. Valores muito baixos podem diminuir a eficácia do algoritmo devido à redução do número de soluções viáveis encontradas nesse menor intervalo, diminuindo, desta forma, as chances de encontrar solução de melhor qualidade entre as soluções candidatas. Valores muito altos, apesar de aumentar as chances de encontrar uma solução de melhor qualidade, podem depreciar a eficiência do AG, causando depreciação no tempo de processamento da solução. A dificuldade em escolher um valor de Δk adequado se deve a uma possível relação desse parâmetro com o tamanho e a complexidade dos problemas, o tamanho de população, dentre outros aspectos do AG. Considerando que neste trabalho este valor foi escolhido subjetivamente, é possível que o valor adotado não tenha sido o mais adequado e, conseqüentemente, tenha depreciado a eficácia e a eficiência do AG. Existem diversos outros critérios de parada, porém eles precisam ser mais bem investigados.

Os resultados do teste L&O para o tamanho de população mostraram que apenas os tamanhos iguais a 75 e 100 não diferiram entre si a 1% de probabilidade. A eficácia do AG para os diferentes tamanhos de população estudados está na Figura 10. Pode-se observar que tamanhos de população iguais a 75 e 100 propiciaram melhores eficácias para o AG. Segundo Youssef et al. (2001), uma das dificuldades que surgem na confecção de um AG é definir o tamanho da população. De acordo com estes autores, uma população pequena pode causar convergência prematura do algoritmo, enquanto uma população muito grande pode demandar muito tempo de processamento para o algoritmo encontrar uma solução boa.

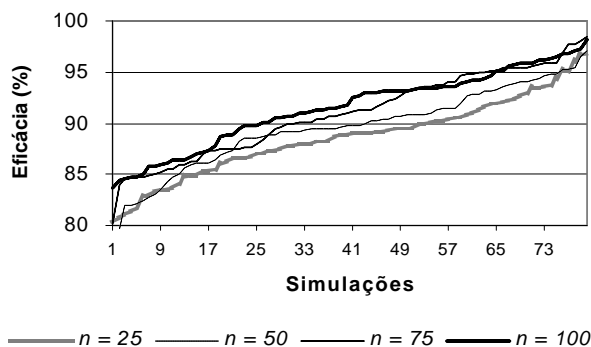


Figura 10 – Medida de eficácia do AG para diferentes tamanho de população nos quatro problemas analisados.
Figure 10 – Efficacy measurements of the AG for different population sizes for the four problems analyzed.



Não houve, também, efeito significativo do tipo de *crossover* na eficácia do AG, de acordo com o teste de L&O, a 1% de probabilidade.

Os valores mínimos, médios e máximos da eficácia do AG, considerando os diferentes métodos de seleção, o tamanho de população e o tipo de *crossover*, foram iguais a 78,95, 90,15 e 98,48%, respectivamente. Nestes casos, a eficácia do AG apresentou um coeficiente de variação igual a 4,51%. Entretanto, ao escolher apenas os melhores métodos de seleção (*Boltzmann* e escalonamento) e o tamanho de população (75 e 100), o AG apresentou eficácia mínima, média e máxima de 90,01, 94,28 e 98,48%, respectivamente (Figura 11). O coeficiente de variação para esta configuração de parâmetros foi igual a 2,08%. Portanto, houve melhoria no desempenho do AG, quando foi feita a melhor escolha de seus parâmetros.

Apesar de a eficiência do AG não ter sido avaliada rigorosamente neste trabalho, o que iria requer análise de complexidade de algoritmos mais sofisticadas, como complexidade de tempo e de espaço, os resultados computacionais preliminares já mostraram vantagens do AG em relação 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 ao AG. Para o problema 4, com 423 variáveis de decisão, o algoritmo *branch and bound* consumiu um tempo cinco vezes superior ao AG (Figura 11).

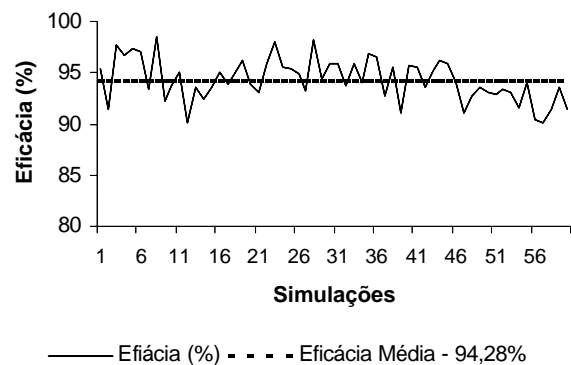


Figura 11 – Medidas de eficácia do AG nos quatro problemas analisados, considerando os métodos de seleção de *Boltzmann* e escalonamento e tamanhos de população iguais a 75 e 100 indivíduos.

Figure 11 – Efficacy measurements of the AG of the four problems analyzed, considering the Boltzmann and assignment selection methods and population sizes equal to 75 and 100 individuals.

As vantagens em termos de tempo de processamento do AG em relação aos métodos exatos de solução podem ser melhoradas através de aperfeiçoamentos na estrutura do algoritmo. Além disto, é necessário aumentar a capacidade do AG para resolver problemas maiores e mais complexos, a fim de que possam ser utilizados para solução de problemas reais de planejamento florestal. Para aumentar a velocidade de processamento do AG é necessário considerar a possibilidade de adotar uma estrutura de dados para otimizar operações como geração das populações e suas respectivas avaliações, como cálculo da *fitness* e testes dos indivíduos, segundo as restrições do problema (avaliação de sua viabilidade). É consenso na literatura especializada que tais operações são críticas e consomem a maioria dos recursos de processamento do AG (Mitchell, 1996).

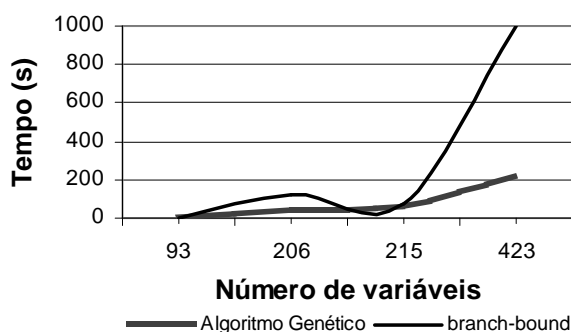


Figura 12 – Comparações de tempo de solução obtida pelo AG e o algoritmo *branch and bound* nos quatro problemas analisados.

Figure 12 – Comparisons of the solution time obtained by the AG and *branch and bound* algorithm in the four problems analyzed.

4. CONCLUSÕES

Os principais objetivos deste trabalho foram desenvolver e testar um algoritmo genético (AG) para resolver problemas de planejamento florestal de natureza combinatorial. Foram testados os efeitos dos métodos de seleção, do tamanho da população e do tipo de *crossover*. Houve diferenças estatísticas significativas a 1% de probabilidade entre os níveis ou categorias de alguns parâmetros avaliados. Os melhores métodos de seleção foram os de *Boltzmann* e escalonamento, e os melhores tamanhos de população foram iguais a 75 e 100 indivíduos. O tipo de *crossover* não afetou a eficácia do AG.

Os melhores parâmetros do AG forneceram, em média, soluções que alcançaram 94,28% do valor ótimo, chegando a atingir valores máximos de 98,48% do ótimo, com um tempo de processamento cinco vezes inferior ao principal algoritmo exato de solução, *branch and bound*. Os resultados mostraram também que os métodos de seleção, *Boltzmann* e escalonamento, foram os que produziram melhores resultados.

A aplicabilidade universal dos AGs, propiciada pela flexibilidade desses algoritmos em resolver problemas de natureza variada, permite concluir que eles podem ser utilizados como alternativa atraente para resolver outros problemas do setor florestal. Alguns exemplos envolvem restrições espaciais, como as restrições de colheita em talhões adjacentes para evitar a abertura de extensas clareiras, o roteamento de veículos para o transporte florestal e diversos outros problemas de natureza combinatorial encontrados no gerenciamento florestal.

O melhor desempenho do algoritmo genético é dependente da escolha de uma boa configuração de seus parâmetros. Como não existem conclusões seguras para a maioria dos problemas de planejamento florestal, em especial os problemas maiores e, ou, mais complexos, essa configuração deve ser escolhida com base 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. 29-301, 1999.
- CERNY, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Application*, v. 45, n. 1, p. 41-51, 1985.
- GARCIA, A. L.; GARCIA, B.; FRIEDMANN, C. V. Resolução de um problema de equilíbrio de trabalho e distribuição de agentes de serviço. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 32., 2000, Viçosa. *Anais...* Viçosa: Universidade Federal de Viçosa, 2000. p. 923-935.

GLOVER, F. Heuristics for integer programming using surrogate constraints. **Decision Science**, v. 8, p. 156-166, 1977.

GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Massachusetts: Addison-Wesley, 1989. 432 p.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Harbor: University of Michigan Press, 1975. 334 p.

IGNIZIO, J. P.; CAVALIER, T. M. **Linear programming**. Englewood Cliffs: Prentice Hall, 1994. 666 p.

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. **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**. v. 33, p. 1105-1118, 2002.

LOCKWOOD, C.; MOORE, T. Harvest scheduling with spatial constraints: a simulated annealing approach. **Canadian Journal Forest Research**, v. 23, p. 468-478, 1993.

MITCHELL, M. **An introduction to genetic algorithms**. London, A Bradford Book, 1996. 205 p.

MURRAY, A. T.; CHURCH, R. L. Analyzing cliques for imposing adjacency restrictions in forest models. **Forest Science**, v. 42, n. 2, p. 45-52, 1996.

NOBRE, S.R. **A heurística da Razão-R aplicada a problemas de gestão florestal**. 1999. 128 f. Dissertação (Mestrado em Administração Rural) – Universidade Federal de Lavras, Lavras, 1999.

REBELLO, F. R.; HUMACHER, S. Uma proposta de algoritmo genético de duas fases para roteamento de veículos. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 32., 2000, Viçosa. **Anais...** Viçosa: Universidade Federal de Viçosa, 2000. p. 672-688.

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: Universidade Federal de Viçosa, 2000. p. 974-995.

WEINTRAUB, A. et al. A heuristic system to solve mixed integer forest planning models. **Operation Research**, v. 42, n. 6, p. 1010-1024, 1994.

YOSHIMOTO, A.; BRODIE, D. Comparative analysis of algorithms to generate adjacency constraints. **Canadian Journal Forest Research**, v. 24, p. 1277-1288, 1993.

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.