

Problema de alocação de áreas de florestas

Marcelo Gonçalves Narciso

Embrapa Informática Agropecuária
C Postal 6041 - 13083-886 – Campinas. SP
E-mail: narciso@cnptia.embrapa.br

Luiz Antonio Nogueira Lorena

INPE – Instituto Nacional de Pesquisas Espaciais
C. P. 515 – 12245-970, São José dos Campos, SP
E-mail: lorena@lac.inpe.br.

Para indústrias que exploram recursos de florestas é interessante manter áreas de reserva ou ainda áreas para reflorestamento. Assim, uma dada floresta pode ser dividida em vários setores para que alguns sejam explorados e outros fiquem de reserva ou sejam recuperados. Deve-se levar em conta também os fatores (restrições) como transporte dos produtos, manutenção da biodiversidade, etc.

Existem modelos sobre o tema levando-se em conta diversas restrições. Em [1], tem-se uma descrição de vários modelos relativos a florestas e exploração destas de forma a preservá-la.

Neste trabalho, foram desenvolvidos algoritmos para resolver um destes problemas usando-se algoritmos genéticos construtivos, aliados a heurísticas que forneceram soluções viáveis e tempo de execução muito bons.

O algoritmo genético construtivo (AGC) [3], aliado a heurísticas construídas para este fim, foi utilizado para fornecer a melhor solução viável possível aos problemas enfocados.

Para um melhor entendimento do que foi feito, nos parágrafos seguintes tem-se uma pequena descrição do funcionamento do AGC.

A teoria tradicional do algoritmo genético (AGT) assume que ele funciona descobrindo, enfatizando e recombinando bons *blocos construtivos* nas soluções. Acredita-se que boas soluções são obtidas através da agregação de bons blocos construtivos, idéia que foi formalizada através da introdução da definição de *esquema* (Holland [2]).

O AGC apresenta algumas características inovadoras em relação aos algoritmos genéticos tradicionais (AGT), tais como população formada apenas de estruturas e/ou esquemas, processo proporcional de avaliação, recombinação entre esquemas, população dinâmica, mutação em estruturas completas, e a possibilidade de uso de heurísticas na representação dos esquemas e/ou estruturas [3].

Em nossa aplicação do AGC ao problema de localização de florestas, cada um

destes é considerado como um problema de *clustering*. Uma representação binária é usada para esquemas e estruturas. Esquemas não levam em conta todos os dados do problema. São recombinados podendo gerar novos esquemas ou estruturas. Novos esquemas são avaliados de forma proporcional e podem entrar na população passarem por um teste de evolução. Quando estruturas completas resultam da recombinação de esquemas ou quando bons esquemas são complementados, as estruturas sofrem mutação. A melhor estrutura gerada é guardada no processo.

No Algoritmo Genético Construtivo os esquemas são avaliados diretamente através das funções de avaliação, o que permite a junção de diferentes *bons* esquemas, o que por sua vez pode dar origem a novos esquemas ou estruturas de alta qualidade. A avaliação direta dos esquemas no AGC representa uma das diferenças em relação ao AGT, onde os esquemas são avaliados através das instâncias que produzem.

O AGC é um algoritmo iterativo e a cada nova iteração do processo, a população aumenta com a inclusão de novos indivíduos (estruturas) gerados com a combinação e mutação de estruturas da população atual. Os bons esquemas (ou estruturas) são preservados dentro dos novos esquemas (ou estruturas) gerados, que geralmente se aproximam de representações completas da solução do problema. Também, a cada iteração, um parâmetro evolutivo é usado na eliminação de indivíduos que não satisfazem um critério de permanência na população. Uma cópia da melhor solução encontrada até o momento é mantida salva, e o processo termina quando um limite de iterações previamente estipulado é atingido ou eventualmente a população fique vazia [3].

O AGC pode ser resumido assim:

AGC() {Algoritmo Genético Construtivo}

$\alpha=0$;

$\epsilon=0.05$; {intervalo de tempo}

Inicializar P_α ; {população inicial}

Avaliar P_α ; {usar as funções f e g}

Para todo s_k (k -ésima estrutura) $\in P_\alpha$ **calcule** $\delta(s_k)$
{cálculo do rank}

Fim_Para

Enquanto não (condição de parada) **fazer**

Para todo $s_k \in P_\alpha$ satisfazendo $\alpha < \delta(s_k)$ **fazer**

{teste de evolução}

$\alpha = \alpha + \epsilon$

Selecionar $P_{\alpha+\epsilon}$ de P_α ; {operador
reprodução}

Recombinar P_α ;

Avaliar P_α ; {calcula adaptação
proporcional, usando f e g}

Fim_Para

Para todo novo $s_k \in P_\alpha$ **calcule** $\delta(s_k)$
{cálculo do rank}

Fim_Para

Fim_enquanto

As funções f e g são limitantes superior e inferior respectivamente associados a cada estrutura s_k ou ainda, podem ser consideradas funções de avaliação de uma estrutura. A função g dará uma solução viável para o problema, respeitando cada esquema gerado e a função f, o limite superior, mas não respeitando todas as restrições. As funções f e g são heurísticas e devem ser apropriadas para cada tipo de problema. Estes algoritmos e o AGC adaptado ao problema de localização de áreas em uma dada floresta são as contribuições deste trabalho. Mais informações sobre o algoritmo do AGC pode ser vistos em [3].

Dos vários problemas existentes, relativos a florestas, enfocamos o problema de reflorestamento de subáreas de uma dada área de florestal. Sejam N o número de subáreas e seja também a restrição de que áreas subjacentes não podem ser desmatadas no mesmo período de tempo ou em períodos de tempo adjacentes. Além disso, em cada subárea deverá ter retirada de madeiras de uma só vez. O objetivo é colher a maior quantidade de madeira possível atendendo estas restrições. Este problema pode ser descrito assim:

Sejam i o índice de cada subárea e t um índice para cada período de tempo. $X_{it} = 1$ se a subárea i é escolhida para a colheita em um dado período (t) e $X_{it} = 0$ se a subárea não for escolhida. Seja V_{it} o presente valor da colheita da subárea i no tempo t. Seja A tal que $A = \{(i,j) \mid \text{subáreas } i \text{ e } j \text{ sejam adjacentes}\}$. A formulação matemática do problema seria:

$$\text{Maximizar } \sum_i \sum_t V_{it} X_{it} \quad (1)$$

$$\text{Sujeito a } X_{it} + X_{jt} \leq 1, \quad (2)$$

$$\sum_t X_{it} \leq 1, \quad \text{para todo } i \quad (3)$$

$$X_{it} = 0 \text{ ou } 1.$$

Os algoritmos para resolver o problema acima, conforme a instância, foram feitos em linguagem C e os testes foram executados em uma estação FreeBSD, CPU 2.66GHz, 2 GB de memória RAM.

Alguns dos resultados obtidos, para algumas instâncias, estão abaixo

i x t	Sup	Sol obtida
10 x 5	200.4	198.6
20 x 5	367.7	365.6
30 x 5	575,8	572,8
40 x 5	723,6	719,5

Tabela 1: Valores obtidos conforme instâncias

Na tabela acima, Sup é o valor superior à solução ótima. Os valores obtidos estão em Sol. Obtida. As instâncias foram geradas pelos autores. Observa-se que os limites obtidos estão muito próximos e os resultados devem ser pelo menos quase ótimos.

Referências

- [1] R L Church et al., Location issues in forest Management. Location Science, 1998.
- [2] HOLLAND, J.H. Adaptation in natural and artificial systems. Massachussets: MIT Press, 1975, p. 11-147.
- [3] L A N Lorena e J C Furtado, Constructive genetic algorithm for clustering problems. *Evolutionary Computation* 9 (3): 309-327, 2001