

UNIVERSIDADE FEDERAL DO PARANÁ

ELDEMAR JASKIU

**PLANEJAMENTO FLORESTAL ESPACIAL ATENDENDO EXIGÊNCIAS DE
ÁREAS MÁXIMAS E MÍNIMAS CONTÍNUAS DE COLHEITA**



CURITIBA

2015

ELDEMAR JASKIU

PLANEJAMENTO FLORESTAL ESPACIAL ATENDENDO EXIGÊNCIAS DE
ÁREAS MÁXIMAS E MÍNIMAS CONTÍNUAS DE COLHEITA

Dissertação apresentada ao curso de Pós-Graduação em Engenharia Florestal do Setor de Ciências Agrárias da Universidade Federal do Paraná como requisito parcial à obtenção do título de Mestre em Ciências Florestais

Orientador: Prof. Dr. Julio Eduardo Arce
Co-orientador: Prof. Dr. Arinei Carlos Lindbeck da Silva

CURITIBA

2015

Ficha catalográfica elaborada pela
Biblioteca de Ciências Florestais e da Madeira - UFPR

Jaskiu, Eldemar

Planejamento florestal espacial atendendo exigências de áreas máximas e mínimas contínuas de colheita / Eldemar Jaskiu. – Curitiba, 2015.
63. : il.

Orientador: Prof. Dr. Julio Eduardo Arce

Coorientador: Prof. Dr. Arinei Carlos Lindbeck da Silva

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Agrárias, Programa de Pós-Graduação em Engenharia Florestal. Defesa: Curitiba, 27/03/2015.

Área de concentração: Manejo Florestal.

1. Florestas - Planejamento. 2. Manejo florestal. 3. Colheita florestal. 4. Teses. I. Arce, Julio Eduardo. II. Silva, Arinei Carlos Lindbeck da. III. Universidade Federal do Paraná, Setor de Ciências Agrárias. IV. Título.


CDD – 634.9

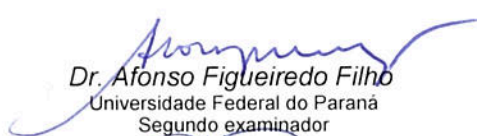
CDU – 634.0.62

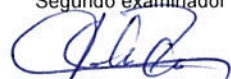
PARECER

Defesa nº. 1112

A banca examinadora, instituída pelo colegiado do Programa de Pós-Graduação em Engenharia Florestal, do Setor de Ciências Agrárias, da Universidade Federal do Paraná, após arguir o(a) mestrando(a) *Eldemar Jaskiu* em relação ao seu trabalho de dissertação intitulado "**PLANEJAMENTO FLORESTAL ESPACIAL ATENDENDO EXIGÊNCIAS DE ÁREAS MÁXIMAS E MÍNIMAS CONTÍNUAS DE COLHEITA**", é de parecer favorável à **APROVAÇÃO** do(a) acadêmico(a), habilitando-o(a) ao título de *Mestre* em Engenharia Florestal, área de concentração em MANEJO FLORESTAL.


Dr. Gustavo Valentim Loch
Universidade Federal do Paraná
Primeiro examinador


Dr. Afonso Figueiredo Filho
Universidade Federal do Paraná
Segundo examinador


Dr. Julio Eduardo Arce
Universidade Federal do Paraná
Orientador e presidente da banca examinadora

Curitiba, 27 de março de 2015.


Antonio Carlos Batista
Coordenador do Curso de Pós-Graduação em Engenharia Florestal



AGRADECIMENTOS

À minha família pelo apoio e incentivo, em especial aos meus pais Lair e Celso por me ensinarem sobre os valores éticos e sobre a importância do conhecimento.

À minha companheira Aline, pela motivação, sonhos compartilhados e por aturar a minha maneira de analisar de modo científico os aspectos da vida.

Aos orientadores Julio e Arinei, que me mostraram o fascinante mundo da pesquisa operacional e me mostraram que é possível transformar muitos problemas da vida real em algo que pode ser resolvido por uma máquina. Isso sem falar na amizade, bom humor e boa receptividade de novas ideias.

A todos os professores que de alguma forma ou de outra contribuíram para a formação do conhecimento necessário para que este trabalho pudesse ser realizado.

À empresa Arauco, pelo apoio com as bases de dados para a aplicação prática do modelo proposto.

Aos meus amigos, colegas de trabalho e colegas do meio acadêmico que de alguma forma contribuíram para a conclusão deste trabalho.

No final das contas a Máquina é apenas uma ferramenta, que pode ajudar a humanidade a progredir mais rápido ao tirar de suas costas o peso dos cálculos e das interpretações. A tarefa do cérebro humano continua sendo a que sempre foi: descobrir novos dados a ser analisados e inventar novos conceitos a ser testados. É uma pena que a Sociedade pela Humanidade não entenda isso.

Isaac Asimov

Eu, Robô "O conflito evitável"

Originalmente publicado em 1950

RESUMO

O planejamento da colheita florestal está entre as etapas mais importantes do empreendimento florestal devido principalmente ao seu alto custo. Planejar espacialmente a colheita está se tornando cada vez mais comum, pois os problemas que eram considerados muito restritivos computacionalmente no início da aplicação da pesquisa operacional no setor florestal tiveram suas formulações reorganizadas de maneira a facilitar a resolução, isso sem mencionar que os recursos computacionais e os programas otimizadores (*solvers*) existentes estão cada vez mais capazes de solucionar problemas maiores e mais complexos. Representar as vizinhanças entre unidades de manejo na forma de restrições tem sido uma tarefa desafiadora, pois cada conjunto de unidades de manejo tem uma conformação diferenciada e para que todas as combinações possíveis entre as unidades fiquem explícitas é necessário um esforço computacional a mais que o esforço de apenas resolver o modelo matemático, em outras palavras, gerar o modelo matemático também é uma tarefa que merece atenção. Neste trabalho foi utilizado um grupo de talhões (unidades de manejo) com apenas 26 unidades e posteriormente foi aplicada a mesma metodologia em uma área comercial com cerca de 10 mil hectares plantados com o gênero *Pinus*. As formulações e algoritmos já existentes para este tipo de problema foram utilizados com sucesso e graças à facilidade que linguagens computacionais matemáticas proporcionam foi possível criar uma rotina que consegue formular as restrições para qualquer floresta, desde que seja fornecida uma matriz de adjacência das unidades de manejo. Adicionalmente também é apresentado um procedimento para automatizar a criação da matriz de adjacência usando uma rotina dentro de um programa de Sistemas de Informações Geográficas (SIG). Os conjuntos de unidades de manejo testados tiveram um decréscimo no Valor Presente Líquido à medida que foram acrescentadas as restrições de áreas máxima e/ou mínima contínuas, assim como um maior tempo necessário para que o *solver* conseguisse encontrar a solução ótima.

Palavras-chave: Adjacência de Talhões; Otimização do Planejamento Florestal; Modelo de Restrição por Área; Blocos de Colheita.

ABSTRACT

The forest harvesting planning is among the most important stages of forest activity because its high costs. Planning spatially the harvest is becoming increasingly common, as problems that were considered too restrictive computationally early in the application of operations research in the forestry sector had their formulations reorganized in order to facilitate the resolution, not to mention that the current computational resources and existing solvers programs are progressively more able to face larger and more complex problems. Represent the neighborhoods between management units in the form of restrictions has been a challenging task, as each set of management units has a different spatial conformation and, in order to achieve all possible combinations among them, an extra computational effort is needed apart than just the effort to solve the problem, in other words, to generate the problem is also a task that deserves attention. In this study it was first used a group of stands (management units) with only 26 units and the same methodology was extended to a commercial area with around 10 thousand hectares planted with *Pinus*. The formulations and algorithms existent for this type of problem have been successfully used and thanks to the aid provided by mathematical computer languages it was possible to create a script that can make restrictions to any forest, as long as an adjacency matrix of the management units is provided. Additionally, it is also presented a procedure to automate the creation of the adjacency matrix using a routine within a Geographic Information Systems program (GIS). The sets of management units tested had a decrease in Net Present Value as the restrictions of maximum and/or minimum continuous area were added, and a longer time was required for the solver to find the optimal solution.

Keywords: Stands Adjacency, Forestry Planning Optimization, Area Restriction Model, Harvest Clusters.

LISTA DE FIGURAS

FIGURA 1 - ESTRUTURA HIERÁRQUICA DO PLANEJAMENTO FLORESTAL	19
FIGURA 2 - COLHEITA DISPERSA, EM BLOCOS E EM BLOCOS DISPERSOS.	24
FIGURA 3 - DISPOSIÇÃO DOS TALHÕES NA ÁREA SIMULADA A SER COLHIDA	25
FIGURA 4 - DISTRIBUIÇÃO DAS ÁREAS EM CADA IDADE	26
FIGURA 5 - PRODUTIVIDADE ESTIMADA PELO SISEUCALIPTO	28
FIGURA 6 - MATRIZ DE ADJACÊNCIA PARA A ÁREA SIMULADA	29
FIGURA 7 - COMPARAÇÃO DO VPL DOS CENÁRIOS.....	35
FIGURA 8 - LOCALIZAÇÃO DA ÁREA DE ESTUDO.....	39
FIGURA 9 - DISTRIBUIÇÃO DAS IDADES DO POVOAMENTO	40
FIGURA 10 - NÚMERO DE UNIDADES DE MANEJO SEGUNDO A QUANTIDADE DE ADJACÊNCIAS.....	44
FIGURA 11 - PERCENTUAL DA ÁREA TOTAL SEGUNDO A QUANTIDADE DE ADJACÊNCIAS.....	44
FIGURA 12 - VARIAÇÃO DO VALOR PRESENTE LÍQUIDO NOS DIFERENTES CENÁRIOS.....	48

LISTA DE QUADROS

QUADRO 1: INFORMAÇÕES DOS TALHÕES.....	27
QUADRO 2: VARIÁVEIS, CONSTANTES E ÍNDICES USADOS NO MODELO MATEMÁTICO	30
QUADRO 3: CENÁRIOS GERADOS PARA A FLORESTA SIMULADA.....	32
QUADRO 4: RESUMO DOS RESULTADOS DOS CENÁRIOS TESTADOS ..	36
QUADRO 5: CENÁRIOS GERADOS PARA A FLORESTA COMERCIAL.....	43
QUADRO 6: RESUMO DOS RESULTADOS DOS CENÁRIOS TESTADOS ..	47

LISTA DE SIGLAS E ABREVIATURAS

AMPL	- <i>A Mathematical Programming Language</i>
ARM	- <i>Area Restriction Model</i>
CBC	- <i>COIN-OR Branch-and-Cut</i>
COIN-OR	- <i>Computational Infrastructure for Operations Research</i>
Embrapa	- Empresa Brasileira de Pesquisa Agropecuária
GIS	- <i>Geographic Information System</i>
GMU	- <i>Generalized Management Unit</i>
MDF	- <i>Medium Density Fiberboard</i>
MDP	- <i>Medium Density Particleboard</i>
NEOS	- <i>Network-Enabled Optimization System</i>
NP-Hard	- <i>Non-Deterministic Polynomial-Time Hard</i>
PI	- Programação Inteira
PIB	- Programação Inteira Binária
RHS	- <i>Right-Hand Side</i>
SIG	- Sistema de Informações Geográficas
URM	- <i>Unit Restriction Model</i>
VPL	- Valor Presente Líquido

Sumário

1. INTRODUÇÃO	11
2. OBJETIVOS	13
2.1. OBJETIVO GERAL	13
2.2. OBJETIVOS ESPECÍFICOS	13
3. REVISÃO DE LITERATURA	14
3.1. PESQUISA OPERACIONAL	14
3.1.1. Programação linear	15
3.2. PLANEJAMENTO ESPACIAL DA COLHEITA FLORESTAL	18
4. FORMULAÇÃO DE UM MODELO DE OTIMIZAÇÃO FLORESTAL ESPACIAL COM RESTRIÇÕES DE ÁREAS MÁXIMAS E/OU MÍNIMAS OPERACIONAIS	25
4.1. OBJETIVOS	25
4.2. MATERIAL E MÉTODOS	25
4.2.1. Área de estudo	25
4.2.2. Restrições de adjacência	28
4.2.3. Modelo de otimização.....	30
4.2.4. Cenários	32
4.2.5. Recursos computacionais utilizados	33
4.3. RESULTADOS E DISCUSSÕES	33
4.4. CONCLUSÕES	37
5. APLICAÇÃO DO MODELO DE OTIMIZAÇÃO FLORESTAL ESPACIAL EM UM POVOAMENTO COMERCIAL.....	38
5.1. OBJETIVOS	38
5.2. MATERIAL E MÉTODOS	38
5.2.1. Área de estudo	38
5.2.2. Restrições de adjacência	40
5.2.3. Horizonte de planejamento.....	41
5.2.4. Modelo de otimização.....	42
5.2.5. Cenários	42
5.2.6. Recursos computacionais utilizados.....	43
5.3. RESULTADOS	43
5.4. CONCLUSÕES	49
REFERÊNCIAS.....	50
ANEXOS	52

1. INTRODUÇÃO

Segundo Bettinger et al. (2009), o manejo de florestas envolve a integração de práticas silviculturais e conceitos econômicos de modo que a floresta atinja os objetivos de seu proprietário da melhor maneira possível. Os autores também discorrem sobre os desafios econômicos relacionados ao manejo de florestas como, por exemplo, a necessidade de obter lucro, trabalhar com um orçamento limitado, gerar entrada de caixa e se manter em um nível competitivo quando comparado com outros investimentos.

Também são apresentados pelos autores diversos objetivos ambientais e sociais como, por exemplo, a manutenção de habitats para o desenvolvimento da vida selvagem, qualidade da água, qualidade do solo, qualidade do ar, biodiversidade e práticas silviculturais como as queimas controladas, que podem afetar a qualidade de vida de moradores próximos às áreas florestais.

Weintraub et al. (2009) reforçam que empresas florestais estão enfrentando o desafio de se adaptar a metodologias de planejamento da colheita que atendam certas restrições ambientais, principalmente medidas de proteção de espécies em seus habitats naturais, com o mínimo de impacto nos ganhos econômicos. Para modelar estes problemas são utilizados modelos explicitamente espaciais para a representação das florestas e dos planos de colheita.

O cronograma de colheita é então modelado como um problema de análise combinatória a ser otimizado, com variáveis de decisão correspondentes aos talhões potenciais de corte em cada momento, onde o objetivo pode consistir em maximizar a receita sujeito a restrições operacionais e ambientais, como por exemplo as restrições de proibição de corte de talhões vizinhos que excedam a área máxima de corte permitida (WEINTRAUB et al., 2009).

Neste trabalho é apresentada uma metodologia para modelar este tipo de problema, assim como os resultados do modelo aplicado à uma área simulada pequena, mas com condições muito próximas aos problemas reais, para auxiliar a apresentação do procedimento, e também os resultados do modelo aplicado à

uma área comercial, para demonstrar a flexibilidade do modelo ao se adaptar para diferentes conformações espaciais.

2. OBJETIVOS

2.1. OBJETIVO GERAL

Apresentar uma metodologia para planejamento espacial da colheita florestal atendendo exigências de áreas máximas e mínimas contínuas.

2.2. OBJETIVOS ESPECÍFICOS

- Desenvolver e apresentar um modelo matemático de programação linear inteira com as restrições de área máxima e mínima contínuas;
- Aplicar o modelo apresentado em um plantio comercial;
- Avaliar o impacto na redução do Valor Presente Líquido que a floresta consegue fornecer sob várias condições de restrições de área máxima e/ou mínima contínua, assim como o impacto no tempo necessário para concluir a busca pela solução ótima do modelo;

3. REVISÃO DE LITERATURA

3.1. PESQUISA OPERACIONAL

Os autores Hillier e Lieberman (2013) introduzem a Pesquisa Operacional mencionando a necessidade de alocação de recursos em organizações: com o crescimento das organizações as decisões gerenciais necessitam priorizar o que é melhor para toda a organização e não necessariamente a melhor decisão para apenas uma instância da organização. Isto normalmente gera objetivos conflitantes para a alocação de recursos e quanto maior a complexidade e especialização da organização mais difíceis se tornam as decisões a serem tomadas para que os recursos sejam utilizados de maneira eficiente. Este tipo de problema e a necessidade de encontrar o melhor caminho para solucioná-lo criaram as condições necessárias para o surgimento da Pesquisa Operacional.

Segundo Hillier e Lieberman (2013) é possível sintetizar a Pesquisa Operacional nas seguintes fases:

- I. Definir o problema de interesse e coletar dados;
- II. Formular um modelo matemático para representar o problema;
- III. Desenvolver um procedimento computacional a fim de derivar soluções para o problema com base no modelo;
- IV. Testar o modelo e aprimorá-lo conforme necessário;
- V. Preparar-se para a aplicação contínua do modelo conforme prescrito pela gerência;
- VI. Implementá-lo.

De uma maneira geral pode-se afirmar que a Pesquisa Operacional tem por objetivo encontrar soluções ótimas ou próximas do ótimo para problemas complexos de tomada de decisão.

Buongiorno e Gilless (2002) mencionaram que as primeiras aplicações de Pesquisa Operacional em problemas de manejo florestal ocorreram no início da década de 1960 e desde então o número de trabalhos publicados cresceu rapidamente. Os autores também destacam a importância da análise econômica juntamente com os modelos criados em Pesquisa Operacional, uma vez que

mesmo que os objetivos sejam ecológicos ou ambientais, a análise econômica é utilizada para comparar custos e/ou benefícios de cada alternativa.

3.1.1. Programação Linear

A programação linear possui um lugar especial dentro da Pesquisa Operacional, sua adaptabilidade a um grande número de problemas e os avanços na resolução de problemas deste tipo fizeram com que ela seja provavelmente o ramo mais conhecido da Pesquisa Operacional.

Hillier e Lieberman (2013) acrescentam que o desenvolvimento da Programação Linear tem sido classificado entre os mais importantes avanços científicos do século XX. Segundo os autores:

“A programação linear usa um modelo matemático para descrever o problema em questão. O adjetivo ‘linear’ significa que todas as funções matemáticas nesse modelo são necessariamente ‘funções lineares’. A palavra ‘programação’, nesse caso, não se refere à programação de computador; ela é essencialmente um sinônimo para planejamento. Portanto a programação linear envolve o planejamento de atividades para obter um resultado ótimo, isto é, um resultado que atinja o melhor objetivo especificado (de acordo com o modelo matemático) entre todas as alternativas viáveis.”

Murty (2003) cita que, entre todos os modelos de otimização, o modelo de programação linear é o mais altamente desenvolvido, tendo como vantagens:

- Serem os mais simples para construir e entender entre os modelos de otimização;
- Seu método de resolução (o algoritmo simplex) é uma extensão do método clássico Gauss-Jordan para resolução de equações lineares; e
- Tem numerosas aplicações em uma grande variedade de áreas.

Cormen et al. (2009) mencionaram que além do algoritmo simplex, apresentado por George Dantzig em 1947, existem mais algoritmos para a resolução de problemas de programação linear, entre eles o *ellipsoid algorithm* desenvolvido por L. G. Khachian em 1979 e os algoritmos de ponto interior (*interior-point algorithm*) introduzidos por N. Karmarkar em 1984.

Murty (2003) também apresenta a forma padrão de problemas de programação linear em notação matricial da seguinte forma:

$$\text{Minimizar } z(x) = c^T x + \alpha$$

$$\text{sujeito à: } Ax = b$$

$$x \geq 0$$

Em que $z(x)$ é a função objetivo, c^T é um vetor com os coeficientes da função objetivo (o T designa a transposição do vetor), x é um vetor com as variáveis de decisão, α é um valor constante, A é a matriz com os coeficientes das restrições e b é um vetor com valores constantes para as restrições também conhecido como RHS (*Right-Hand Side*).

A forma padrão apresentada é a configuração que o problema deve apresentar para a aplicação de métodos de resolução. Muitos problemas não apresentam a forma padrão quando são enviados aos programas otimizadores ou *solvers*. Isto acontece pois normalmente estes *solvers* possuem rotinas internas que completam ou reorganizam o problema para que ele assuma a forma padrão, permitindo que o modelo matemático seja escrito de uma forma mais legível à quem irá modelar o problema.

3.1.1.1. Programação Linear Inteira e Programação Linear Inteira Mista

De acordo com Hillier e Lieberman (2013) a Programação Linear Inteira (muitas vezes abreviada por apenas Programação Inteira) constitui-se de um tipo especial de modelos de Programação Linear que possuem uma restrição adicional de que as variáveis devem possuir valores inteiros. Se apenas algumas variáveis possuírem a restrição de valores inteiros o modelo é então denominado de Programação Linear Inteira Mista. Exemplos de problemas com variáveis inteiras são problemas de alocação de pessoas a trabalhos específicos: não é possível atribuir meia pessoa para dirigir um caminhão e a outra metade desta pessoa para dirigir um trator.

Problemas de Programação Linear Inteira e Programação Linear Inteira Mista necessitam de algoritmos mais complexos para serem resolvidos, portanto, a busca pela solução ótima exige um esforço computacional maior, muitas vezes inviabilizando a busca pela solução ótima (casos em que o tempo para gerar a solução constitui uma restrição). Cormen et al. (2009) demonstraram que apenas para determinar se um problema com variáveis inteiras possui ou não uma única solução factível é necessário um algoritmo com tempo de execução classificado como NP-Hard, o que significa que não existe um algoritmo conhecido que consiga resolver o problema em tempo polinomial¹.

3.1.1.2. Programação Linear Inteira Binária ou Programação Inteira

A Programação Linear Inteira Binária, por sua vez, possui uma restrição para que as suas variáveis assumam apenas os valores de zero ou um indicando, por exemplo, uma decisão de sim ou não (verdadeiro ou falso). Estes tipos de restrições aumentam a capacidade do modelo de se adaptar a situações mais próximas da realidade. Hillier e Lieberman (2013) apresentaram situações onde os problemas com variáveis binárias auxiliares podem possuir restrições mutuamente exclusivas ou então grupos de restrições onde apenas um subconjunto destas restrições devem ser obrigatoriamente válidas.

Hillier e Lieberman (2009) exemplificaram a complexidade dos problemas de programação linear inteira:

“Pode parecer que os problemas de PI sejam relativamente fáceis de serem resolvidos. Afinal de contas, os problemas de programação linear podem ser resolvidos de modo extremamente eficiente e, a única diferença é que os problemas de PI têm um número bem menor de soluções a ser consideradas.

(...)

Números finitos podem ser astronomicamente grandes. Considere, por exemplo, o caso simples dos problemas de PIB. Com ‘n’ variáveis há 2^n soluções a serem consideradas (...). Portanto, cada vez que ‘n’ for incrementado em 1, o número de soluções dobra.”

Para a resolução de problemas com variáveis inteiras ou binárias é natural considerar o emprego de algum tipo de procedimento de enumeração (HILLIER

¹ Um algoritmo com tempo polinomial possui tempo de resolução descrito por uma expressão polinomial dependente da entrada (*input*) fornecida.

e LIEBERMAN; 2009). Como exemplo estes autores exemplificam a técnica da ramificação e avaliação progressiva (*branch-and-bound*), que usa como conceito básico a estratégia “dividir para conquistar”:

“Já que o ‘grande’ problema original é muito difícil de ser resolvido diretamente, ele é dividido em subproblemas cada vez menores até que sejam vencidos. A divisão é realizada dividindo-se o conjunto inteiro de soluções viáveis em subconjuntos cada vez menores. A conquista é feita parcialmente, limitando-se quão boa pode ser a melhor solução no subconjunto e, a seguir, descartando o subconjunto caso seu limite indique que, possivelmente, ele não possa conter uma solução ótima para o problema original. ”

3.2. PLANEJAMENTO ESPACIAL DA COLHEITA FLORESTAL

Segundo Buongiorno e Gilles (1987) o manejo florestal é a arte e a ciência de tomada de decisões com relações ao uso e conservação florestal de uma organização. Tais decisões podem envolver um planejamento de longo prazo para vários anos assim como atividades diárias.

O planejamento florestal normalmente é realizado de forma a obter um máximo de retorno econômico de povoamentos florestais respeitando os fatores sociais e ambientais que esta atividade atinge dentro do horizonte de planejamento. A colheita florestal é parte deste planejamento e é uma das etapas mais dispendiosas do setor florestal. Em grandes áreas florestais os povoamentos estão divididos em um número de talhões que não permitem uma rápida decisão sobre qual é o melhor agrupamento de áreas para serem colhidas numa mesma época, sendo conveniente, portanto, o uso de métodos computacionais para alcançar a otimização deste agrupamento, uma vez que se a área destinada a corte for muito pequena o deslocamento da frente de colheita até esta área não será compensatório e se a área for muito grande a limitação poderá ser baseada em aspectos ambientais ou para atender a exigências legais que regulamentam a extensão da floresta que pode ser colhida no período.

Mitchell (2004) apresentou uma divisão hierárquica do planejamento florestal dividida nas seguintes fases:

- O planejamento **estratégico** é o mais longo, com um Horizonte de Planejamento que pode durar de duas a quatro rotações², usado para determinar a sustentabilidade a longo prazo da floresta, assim como questões de uso da terra e tipo de regime silvicultural. Os resultados deste planejamento indicam os volumes a serem colhidos nos primeiros anos sem indicar precisamente quais talhões serão efetivamente colhidos.
- O planejamento **tático** possui um horizonte de planejamento de dois a cinco anos, usa os resultados do planejamento estratégico para determinar quais talhões serão efetivamente colhidos nos próximos anos e pode incluir restrições ambientais e restrições de construção de estradas.
- O planejamento **operacional**, por sua vez, direciona as vendas dos diferentes produtos gerados pela colheita florestal.

O autor também apresentou uma representação gráfica desta hierarquia. (FIGURA 1)

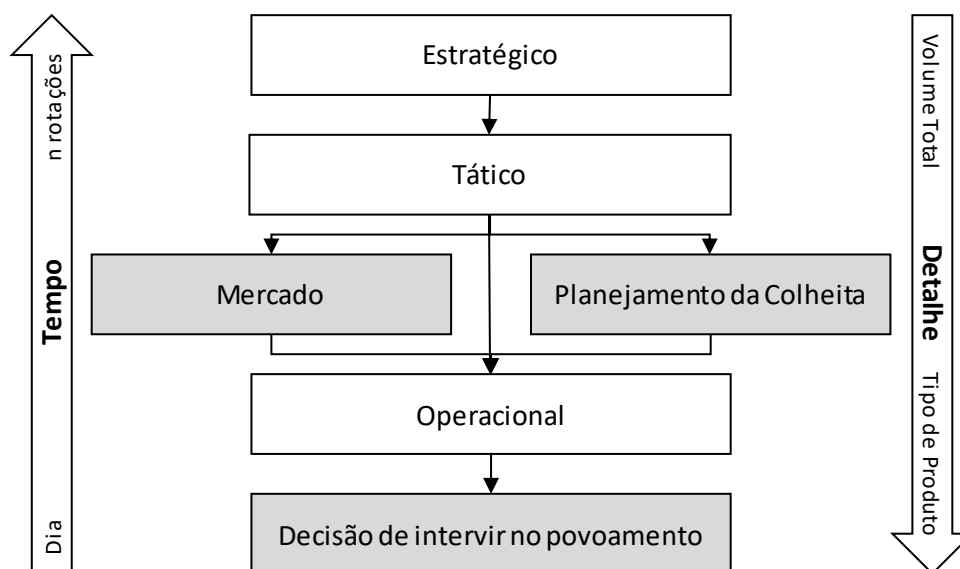


FIGURA 1 - ESTRUTURA HIERÁRQUICA DO PLANEJAMENTO FLORESTAL

Fonte: adaptado de Mitchell (2004).

² Uma rotação é o intervalo de tempo entre o plantio e a colheita da floresta.

O planejamento florestal espacial vem evoluindo nos últimos anos e o planejamento com base em adjacências de talhões consiste em uma metodologia que está se tornando cada vez mais comum. A exemplo o incentivo a processos de sustentabilidade dado pela *American Forest and Paper Association* nas florestas nos Estados Unidos, ao restringir o corte de áreas contínuas com mais de 120 acres (48,55 ha) (GOYCOOLEA et al., 2005).

Nascimento (2015) menciona que Johnson e Scheurman³ (1977) classificaram os modelos de planejamento florestal em dois tipos: o modelo tipo I e o modelo tipo II, que são diferenciados por basicamente quatro características:

- O modelo tipo I não altera as áreas das unidades de manejo ao longo de todo o horizonte de planejamento;
- O modelo tipo II permite que as áreas sejam alteradas ao longo do horizonte de planejamento, agrupando ou dividindo unidades de manejo em novas unidades em cada momento de corte raso e plantio da unidade de manejo;
- É necessária uma restrição de área para cada unidade de manejo em ambos os tipos;
- No modelo tipo I, para cada unidade de manejo considerada no primeiro período, uma atividade deve ser definida para cada sequência possível de corte ao longo do horizonte de planejamento, enquanto que no modelo tipo II, para cada classe de idade do primeiro período, uma atividade deve ser definida para cada período até que esta classe seja colhida ou deixada em estoque de produção ao final do horizonte de planejamento. No modelo tipo II, para cada período, uma atividade também deve ser definida para cada possível duração de tempo até a área ser novamente colhida ou deixada em estoque no fim do horizonte de planejamento.

³ JOHNSON, K. N., SCHEURMAN, H. L. **Techniques for prescribing optimal timber harvest and investment under different objectives - discussion and synthesis.** Forest Science, Monograph 18, p.1-31, 1977.

Gomide (2009) mencionou que os debates em prol do controle da colheita em extensas áreas de florestas contribuíram para o surgimento do conceito de proteção de habitats, conhecido como *green up*. O mesmo autor explica que, segundo McNaughton et al. (2001), o efeito aplicado desta medida promove um processo de dispersão da colheita florestal na área, durante a execução da colheita. Esta dispersão é explicada quando um talhão é selecionado para o corte e, pela regra, seus vizinhos ficam impedidos de ser colhidos durante um determinado período de tempo (*green up delay*).

Segundo Murray (1999), os modelos de restrições por adjacência apresentam variações ligadas a restrições por unidade e por área, sendo então formulados com concepções matemáticas diferentes. O autor propõe uma classificação dos modelos com base no tipo de restrições utilizadas: URM (*Unit Restriction Model*) e ARM (*Area Restriction Model*).

Os modelos que apresentam restrições por unidade (URM) incompatibilizam a colheita florestal de dois ou mais talhões vizinhos (STEIGUER et al., 2003⁴ apud GOMIDE, 2009). A outra opção de restrição (ARM) permite a colheita de talhões vizinhos, desde que não ultrapassem um limite máximo de área contínua pré-estabelecida (STEIGUER et al., 2003 apud GOMIDE, 2009). A diferença entre os modelos ARM e URM é o fato de haver a possibilidade de colher talhões vizinhos (respeitando um limite máximo de área) no primeiro caso, e para o segundo caso, não. Além disso, o grau de dificuldade de se formular um modelo ARM é muito maior que no modelo URM (ZHU e BETTINGER, 2007 apud GOMIDE, 2009).

Gomide (2009) também cita que a primeira tentativa de se modelar problemas de adjacência utilizava restrições do tipo pares (*Pairwise*), cuja formulação matemática é considerada mais fácil de aplicar, e talvez por isso mais difundida entre os planejadores florestais. Na classificação de Murray (1999) estas eram restrições do modelo tipo URM. A desvantagem deste tipo de restrição está associada ao elevado número de inequações (restrições)

⁴ STEIGUER, J. E. LIBERTI, L.; SCHULER, A. HANSEN, B. **Multicriteria decision models for forestry and natural resources management: an annotated bibliography**. General Technical Report. NE-307. Newtown Square, PA: U.S. Department of Agriculture, Forest Service, Northeastern Research Station. P.32. 2003.

desenvolvidas para retratar todo o problema (SILVA, 2004⁵ apud GOMIDE, 2009).

O modelo ARM possibilita a colheita de talhões vizinhos, desde que o somatório de suas áreas não ultrapasse a área máxima permitida. Este tipo de restrição espacial é empregado quando não se deseja que o somatório das áreas de talhões vizinhos seja superior ao limite máximo de área permitida na colheita florestal. Esta flexibilidade não é observada no modelo URM (GOMIDE, 2009).

Segundo Murray (1999), o desafio é encontrar uma maneira de formular linearmente este tipo de restrição, pois devido a características não lineares, a sua formulação torna-se impossível via programação linear inteira. A alternativa para os métodos exatos é o uso de procedimentos de aproximação, como as metaheurísticas que, apesar do grau de dificuldade na implementação, permitem encontrar soluções viáveis a partir de funções de penalidade.

Gomide (2009) menciona que Zhu et al.⁶ (2007) consideram o uso de restrições ARM como muito difíceis de serem formuladas via programação linear inteira, reforçando o postulado de Murray (1999), segundo o qual é praticamente impossível formular estas restrições a priori, pois existem muitas combinações possíveis entre talhões vizinhos, tornando o processo bastante exaustivo e passível de erro. A partir deste desafio lançado, alguns pesquisadores trabalharam o tema, e assim propuseram uma metodologia capaz de formular linearmente este tipo de restrição. A proposta se consolidou com o desenvolvimento de um algoritmo que estabelecia as restrições de adjacência, chamado de *Path Algorithm* (McDill et al., 2002). Os autores também desenvolveram outro procedimento de formulação, denominada de GMU (*Generalized Management Unit*). Um ano mais tarde, outro método de formulação foi desenvolvido por Crowe et al. (2003). Nestes exemplos, os autores utilizam a formulação matemática dos modelos, escrita por seus algoritmos, e aplicaram o método de resolução exato via *branch-and-bound*.

⁵ SILVA, R. T. **Planejamento florestal, modelos de programação inteira, multiobjectivo e aplicações**. Coimbra. 2004. 120f. Dissertação de Mestrado, Universidade de Coimbra.

⁶ ZHU, J. BETTINGER, P.; LI, R. **Additional Insight into the Performance of a New Heuristic for Solving Spatially Constrained Forest Planning Problems**. *Silva Fennica* 41(4): 687–698, 2007.

A FIGURA 2 apresenta uma ilustração do problema de adjacências: uma colheita dispersa pode ser resultado de um planejamento onde a área máxima contínua é pequena suficiente para impedir que mais de um talhão seja cortado ao mesmo tempo. A colheita em blocos é resultado da aplicação de restrições de área mínima contínua suficientemente grande para a concentração da colheita de vários talhões em um mesmo período. A colheita em blocos dispersos representa um agrupamento proporcionado pelas restrições de área mínima contínua e, ao mesmo tempo, uma dispersão que é resultado das restrições de área máxima contínua.

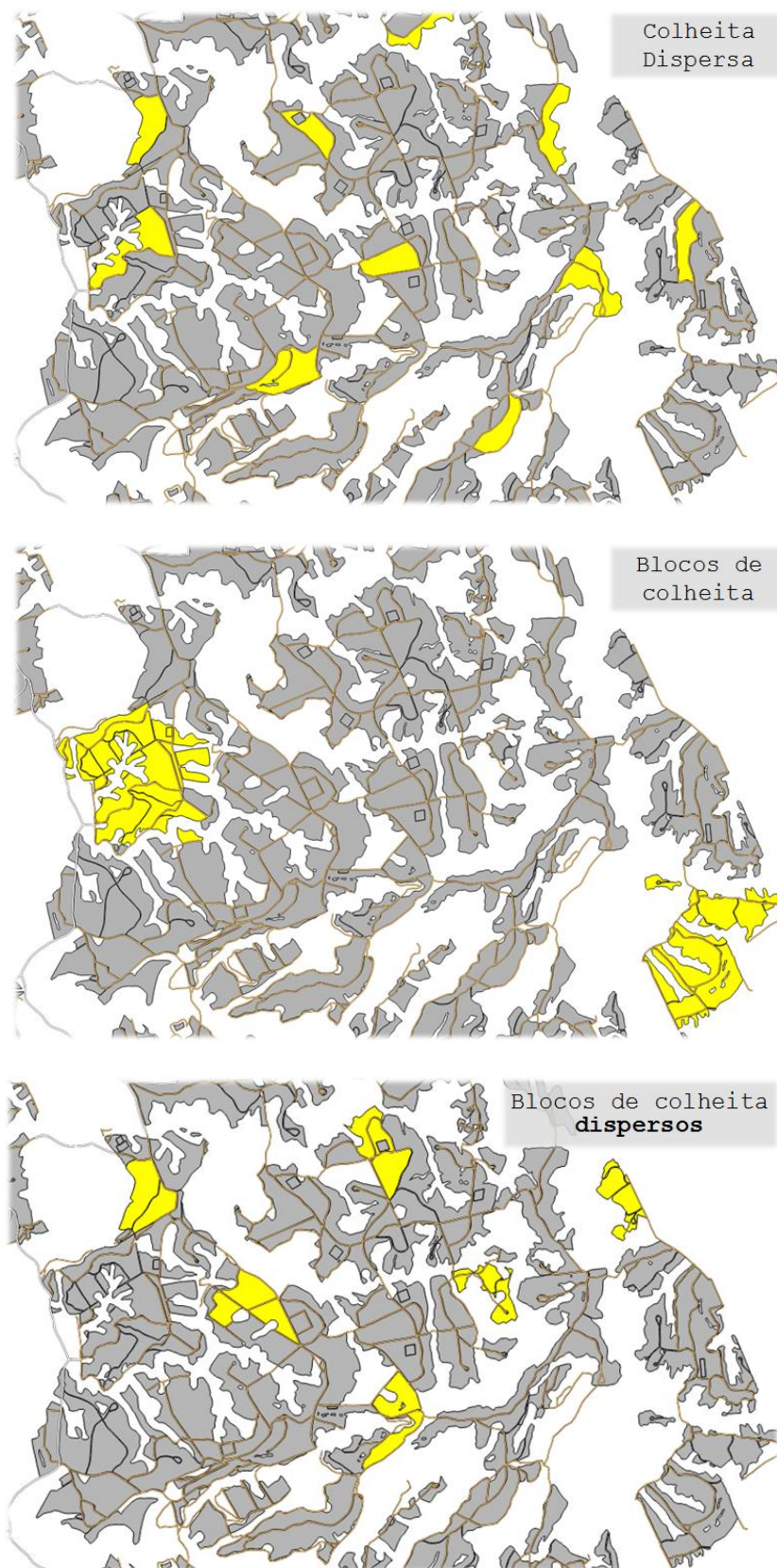


FIGURA 2 - COLHEITA DISPERSA, EM BLOCOS E EM BLOCOS DISPERSOS.
Fonte: O autor (2015)

4. FORMULAÇÃO DE UM MODELO DE OTIMIZAÇÃO FLORESTAL ESPACIAL COM RESTRIÇÕES DE ÁREAS MÁXIMAS E/OU MÍNIMAS OPERACIONAIS

4.1. OBJETIVOS

- Apresentar o modelo matemático de programação linear inteira com as restrições de áreas máximas e mínimas contínuas operacionais;
- Avaliar o impacto na redução do Valor Presente Líquido que a floresta apresenta sob várias condições de restrições de áreas contínuas máximas e/ou mínimas, assim como o impacto no tempo necessário para concluir a busca pela solução ótima do modelo.

4.2. MATERIAL E MÉTODOS

4.2.1. Área de Estudo

Como exemplo será utilizada uma área hipotética com 26 talhões e área total de 396 hectares, conforme representado na FIGURA 3.

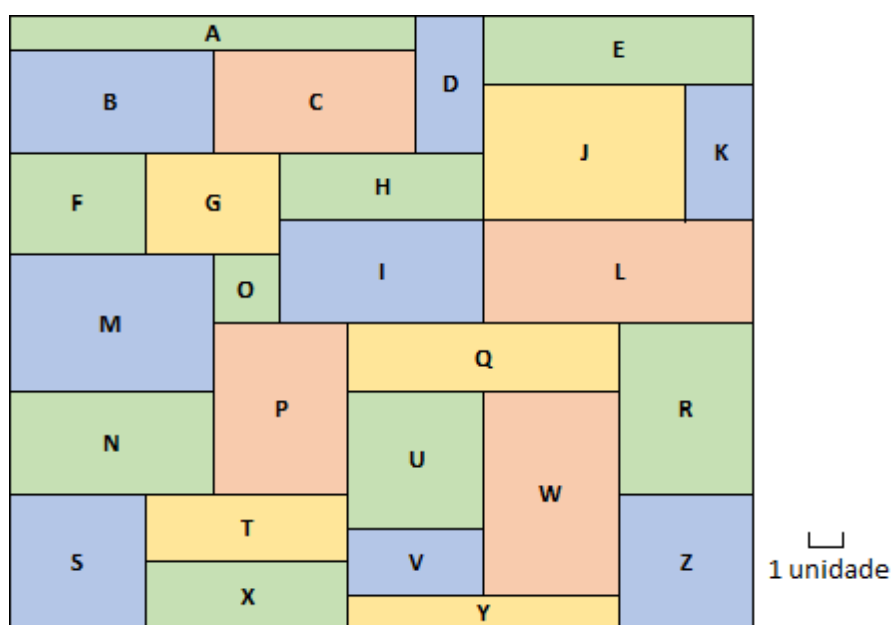


FIGURA 3 - DISPOSIÇÃO DOS TALHÕES NA ÁREA SIMULADA A SER COLHIDA

Fonte: O autor (2015)

As áreas foram definidas proporcionalmente ao tamanho de cada talhão representado na FIGURA 3. A escala da figura foi adotada como sendo uma unidade igual a 100 metros, resultando em 1 unidade² igual a 1 hectare. Assim sendo a área do menor talhão foi de 4 hectares e a área do maior talhão foi 24 hectares.

Foram atribuídas idades aos talhões variando de 1 a 6 anos e para a determinação de qual idade seria atribuída a cada talhão foram realizados vários sorteios e foi escolhido o sorteio que possuía a menor diferença entre a soma de áreas em cada idade, de forma a obter uma floresta mais regulada⁷, conforme representado na FIGURA 4. No QUADRO 1 estão as áreas, idades e adjacências de cada talhão.

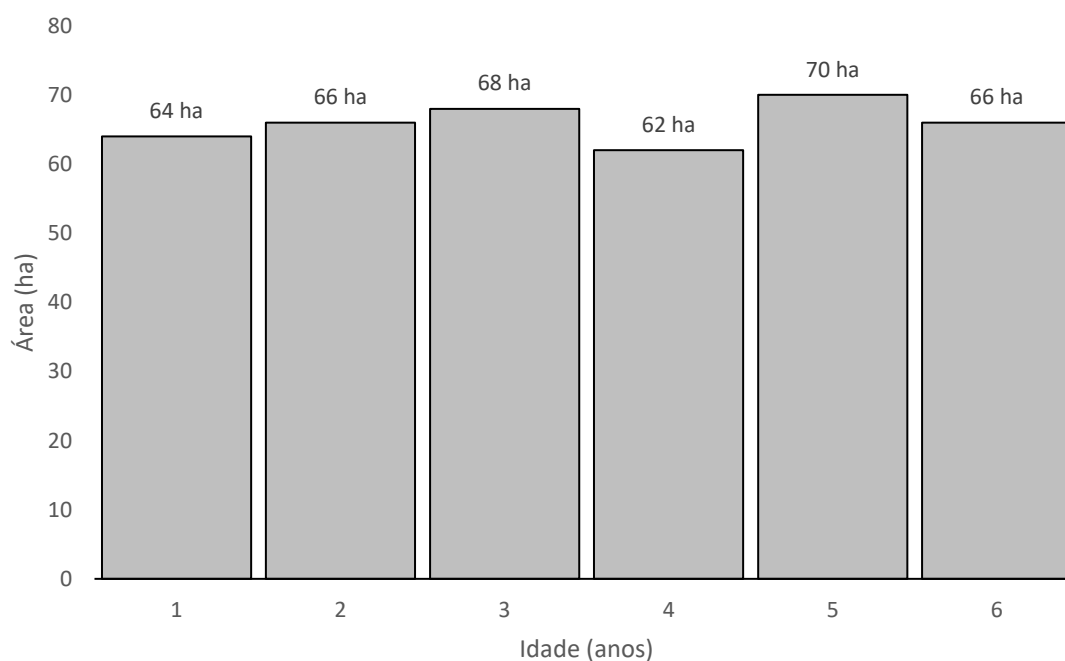


FIGURA 4 - DISTRIBUIÇÃO DAS ÁREAS EM CADA IDADE

Fonte: O autor (2015)

⁷ Neste contexto uma floresta perfeitamente regulada seria uma floresta com exatamente a mesma área sendo plantada e colhida em cada ano, ou seja, cada idade teria exatamente a mesma área plantada (a distribuição de área por idade seria uma função constante).

QUADRO 1 - INFORMAÇÕES DOS TALHÕES

Talhão	Área (hectares)	Idade (anos)	Lista de talhões adjacentes
A	12	3	B, C, D
B	18	2	A, C, F, G
C	18	5	A, B, D, G, H
D	8	1	A, C, E, H, J
E	16	2	D, J, K
F	12	3	B, G, M
G	12	3	B, C, F, H, I, M, O
H	12	5	C, D, G, I, J
I	18	6	G, H, L, O, P, Q
J	24	4	D, E, H, K, L
K	8	3	E, J, L
L	24	3	I, J, K, Q, R
M	24	5	F, G, N, O, P
N	18	4	M, P, S, T
O	4	2	G, I, M, P
P	20	1	I, M, N, O, Q, T, U
Q	16	1	I, L, P, R, U, W
R	20	4	L, Q, W, Z
S	16	5	N, T, X
T	12	1	N, P, S, U, V, X
U	16	2	P, Q, T, V, W
V	8	1	T, U, W, X, Y
W	24	6	Q, R, U, V, Y, Z
X	12	2	S, T, V, Y
Y	8	6	V, W, X, Z
Z	16	6	R, W, Y

Fonte: O autor (2015)

Para a estimativa da produção da floresta (FIGURA 5) foi utilizada uma simulação no programa SisEucalipto, disponibilizado pela Empresa Brasileira de Pesquisa Agropecuária – Embrapa. Os parâmetros utilizados no programa foram Índice de Sítio 33 m, densidade de 1667 árvores por hectare, percentagem inicial de sobrevivência de 99% (versão do *software*: 2008).

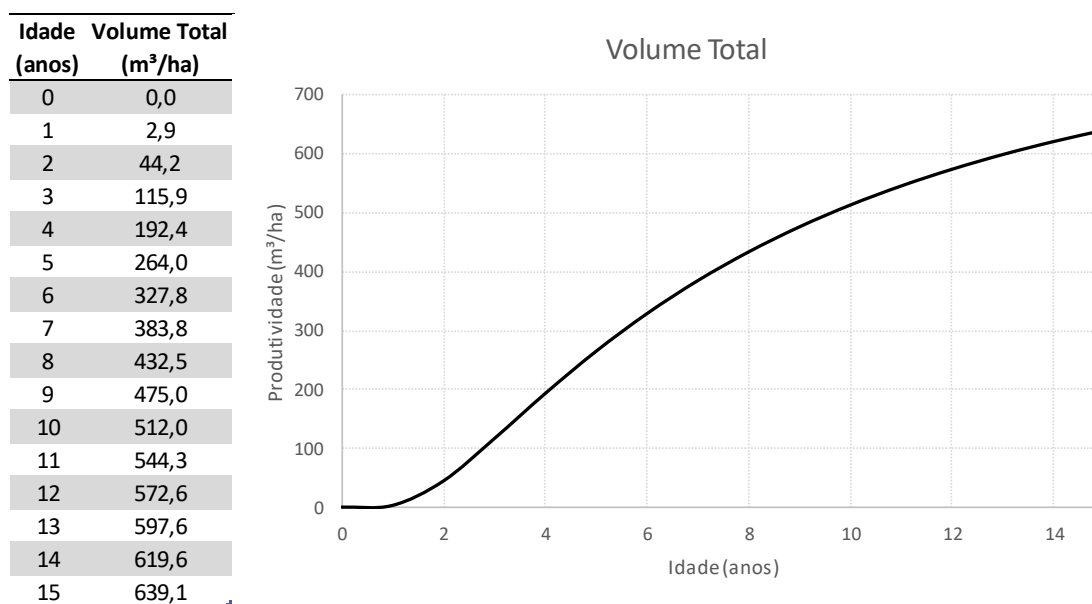


FIGURA 5 - PRODUTIVIDADE ESTIMADA PELO SISEUCALIPTO

Fonte: O autor (2015)

O preço da madeira foi assumido como sendo de R\$ 100,00 m⁻³ e a taxa de desconto aplicada foi de 10% ao ano.

4.2.2. Restrições de Adjacência

O modelo criado em AMPL⁸ utiliza como entrada uma matriz de adjacência, caracterizada por ser uma matriz quadrada e simétrica, onde o número de linhas e colunas é igual ao número de talhões, esta matriz possui valores binários, onde o número zero indica a ausência de adjacência entre os talhões e o número um indica que os talhões são adjacentes. A matriz de adjacência para os talhões aqui apresentados está ilustrada na FIGURA 6.

⁸ AMPL: *A Mathematical Programming Language*, é uma linguagem de programação que permite escrever o conjunto de equações de um modelo matemático para posterior interpretação de um otimizador (*solver*).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
A	1	1	1	1																							
B	1	1	1			1	1																				
C	1	1	1	1				1	1																		
D	1			1	1	1			1		1																
E				1	1					1	1																
F		1				1	1						1														
G		1	1			1	1	1	1				1		1												
H			1	1				1	1	1																	
I								1	1	1			1			1	1	1									
J				1	1				1	1	1																
K											1	1	1														
L										1	1	1	1						1	1							
M						1	1						1	1	1	1											
N													1	1		1					1	1					
O							1						1	1	1	1											
P										1	1	1	1	1	1	1						1	1				
Q										1						1	1	1				1	1		1		
R												1															1
S																1					1	1				1	
T																	1	1			1	1	1	1		1	
U																	1	1			1	1	1	1		1	
V																						1	1	1	1	1	1
W																						1	1		1	1	1
X																						1	1		1	1	
Y																							1	1	1	1	1
Z																							1		1	1	1

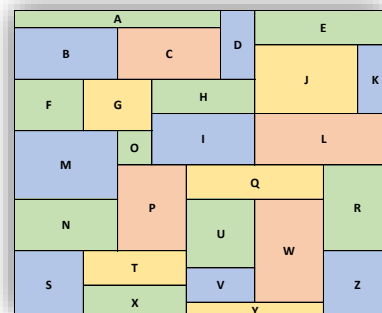


FIGURA 6 - MATRIZ DE ADJACÊNCIA PARA A ÁREA SIMULADA

Fonte: O autor (2015)

As restrições de adjacências são todas as combinações de conjuntos de talhões que ultrapassam a área máxima (não permitindo a colheita de todos este conjunto) ou um agrupamento mínimo de talhões para garantir que a área mínima seja atingida. As combinações foram realizadas pela própria linguagem de modelagem matemática (AMPL).

Foram testadas restrições de área mínima contínua variando de 20 a 40 hectares e áreas máximas contínuas variando de 10 a 65 hectares (o cenário com 10 hectares de área máxima não possui restrições de área mínima). Como pode ser vista no quadro 2.

Quando a área máxima permitida de corte contínua é inferior à área de um único talhão a restrição é ignorada, possibilitando que um talhão com 24 hectares seja cortado quando a restrição não permite que blocos com mais de 20 hectares sejam formados, desde que nenhum talhão adjacente à este seja colhido no mesmo momento. Caso isto se torne um problema é necessário realizar um retalhonamento, de forma que a área do maior talhão seja inferior ao limite de área máxima contínua.

4.2.3. Modelo de Otimização

QUADRO 2 - VARIÁVEIS, CONSTANTES E ÍNDICES USADOS NO MODELO MATEMÁTICO

Variável, Constante ou Índice	Descrição
V	Valor Presente Líquido em R\$
i	Talhão i (ou unidade de manejo i)
I	Número de talhões
j	Ano do horizonte de planejamento
J	Duração do horizonte de planejamento
$X_{i,j}$	Variável binária que representa o corte ou não do talhão i no ano j
A_i	Área do talhão i em hectares
$P_{i,j}$	Produtividade do talhão i no ano j em m^3/ha
M	Preço da madeira em R\$/ m^3
T	Taxa de desconto (%)
$D_{min,j}$	Demanda mínima de madeira no ano j em m^3
C_i	Classe de idade do talhão i no ano zero ⁹
$C_{mín}$	Classe de idade mínima para o corte raso
$E\%$	Variação percentual permitida na produção (em comparação com a produção do ano zero)
L_i	Lista de talhões adjacentes ao talhão i
h	Talhão h . A notação h é utilizada para designar os talhões que estão presentes na lista L_i
$A_{mín}$	Área mínima de corte contínua
$mín(a; b)$	Função que retorna o menor valor entre a e b
K_p	K é uma lista de listas. Cada elemento p da lista K é uma lista. Cada elemento p é uma lista com um conjunto de talhões adjacentes cuja soma das áreas ultrapassa a área máxima permitida

⁹ A classe de idade é o arredondamento da idade para o número inteiro mais próximo.

$card(a)$	Função que retorna o número de elementos da lista a (cardinalidade da lista)
-----------	---

4.2.3.1. Função Objetivo

O modelo de otimização aplicado teve como objetivo a maximização do Valor Presente Líquido, a equação que expressa a função objetivo é apresentada na equação 4.1.

$$Max V = \sum_{i=1}^I \sum_{j=0}^J \left[X_{ij} A_i P_{ij} \frac{M}{(1+T)^j} \right] \quad (4.1)$$

4.2.3.2. Restrições

- Cortar o talhão apenas uma vez durante o Horizonte de Planejamento:

$$\sum_{j=0}^J [X_{ij}] \leq 1 \quad \forall_i \quad (4.2)$$

- Suprir a demanda mínima anual:

$$\sum_{i=1}^I [X_{ij} A_i P_{ij}] \geq Dmin_j \quad \forall_j \quad (4.3)$$

- Respeitar a Idade Mínima de Corte:

$$X_{ij} (C_i + j) \geq X_{ij} C_{min} \quad \forall_i \forall_j \quad (4.4)$$

- Manter o equilíbrio da produção (positivo e negativo):

$$\sum_{i=1}^I [X_{ij} A_i P_{ij}] \leq (1 + E\%) \sum_{i=1}^I [X_{i0} A_i P_{i0}] \quad \forall_j | j > 0 \quad (4.5)$$

$$\sum_{i=1}^I [X_{ij}A_iP_{ij}] \geq (1 - E\%) \sum_{i=1}^I [X_{i0}A_iP_{i0}] \quad \forall_j | j > 0 \quad (4.6)$$

- Área mínima de corte contínua

$$\sum_{h \in L_i} [X_{hj}A_h] \geq \min \left\{ A_{\min}; \sum_{h \in L_i} [A_h] \right\} * X_{hj} \quad \forall_i \forall_j \quad (4.7)$$

- Área máxima de corte contínua

$$\sum_{i \in K_p} [X_{ij}] \leq \text{card}(K_p) - 1 \quad \forall_p \forall_j \quad (4.8)$$

4.2.4. Cenários

Foram criados os cenários apresentados no QUADRO 3 para representar diferentes opções de planejamento, cada um com um diferente conjunto de restrições e/ou um valor diferente para alguma restrição (*RHS*¹⁰). O primeiro cenário não possui restrições de adjacência ativas e foi estabelecido como o cenário “testemunha”, uma vez que por ser menos restritivo apresenta o maior Valor Presente Líquido entre os cenários.

QUADRO 3 - CENÁRIOS GERADOS PARA A FLORESTA SIMULADA

Cenário	Equilíbrio (%)	Demanda (mil m ³)	Idade mínima (anos)	Área Mínima (ha)	Área Máxima (ha)
1	±10	20	5	-	-
2	±10	20	5	20	-
3	±10	20	5	30	-
4	±10	20	5	40	-
5	±10	20	4	40	-
6	±10	-	-	50	-
7	±10	20	5	-	10
8	±10	20	5	-	20
9	±10	20	5	-	30
10	±10	20	5	-	40

¹⁰ *Right-Hand Side*: usado para representar todos os valores que estão à direita do conjunto de equações das restrições.

11	±10	20	5	-	50
12	±10	20	5	-	65
13	±10	20	5	20	40

Fonte: O autor (2015)

4.2.5. Recursos computacionais utilizados

Os problemas foram gerados em uma planilha eletrônica com o auxílio do programa Microsoft Office Excel 2013. O modelo matemático foi escrito em linguagem AMPL utilizando-se para tal o suplemento do Excel chamado SolverStudio (MASON, 2013). O otimizador utilizado foi o CBC (*COIN-OR branch and cut*), desenvolvido pela COIN-OR (*Computational Infrastructure for Operations Research*).

O *hardware* utilizado para a resolução do modelo gerado foi um servidor com dois processadores Intel Xeon X5660 @ 2.8GHz, com 64 GB de memória RAM. Um dos servidores NEOS, localizados na Universidade de Wisconsin, com a versão completa da AMPL instalada e restrito a problemas com até 3GB de tamanho ou oito horas de resolução.

4.3. RESULTADOS E DISCUSSÕES

Todos os cenários gerados têm um tempo de resolução inferior a 3 minutos.

O cenário 1 consegue atender (possui resposta factível) às restrições de demanda, equilíbrio da produção e idade mínima de corte e gera um Valor Presente Líquido de 12,2 milhões de reais.

Os cenários 2 e 3, que além das restrições do cenário 1 possuem restrições de área mínima contínua de 20 e 30 hectares, respectivamente, fazem com que o VPL tenha redução de 1% e 2,5%, enquanto que o tempo de resolução aumenta para 8 e 39 segundos, respectivamente.

O cenário 4, com restrição de área mínima contínua de 40 hectares e as mesmas restrições do cenário 1 é infactível. Para tornar a resolução desta restrição possível foi criado o cenário 5, com um relaxamento na restrição de

idade mínima de corte, possibilitando que talhões com 4 anos pudessem ser cortados. Assim o VPL fica 6,7% menor que o cenário 1 e o tempo de resolução é de 13 segundos.

O cenário 6 com área mínima de 50 hectares não é factível, mesmo removendo a restrição de idade mínima de corte, de demanda mínima anual e relaxando a restrição de equilíbrio da produção para variações de até 50% para mais ou para menos em relação à produção do primeiro ano.

O cenário 7 teve as mesmas restrições do cenário 1 e as restrições de área máxima de corte menor ou igual a 10 hectares. Unidades de manejo únicas com área maior que a área máxima podem ser cortadas, desde que suas adjacentes não tenham corte no mesmo ano. Esta restrição foi equivalente a fazer com que nenhuma unidade de manejo adjacente seja cortada no mesmo ano do horizonte de planejamento, promovendo a máxima dispersão da colheita. O VPL é 1% inferior ao cenário 1, enquanto que o tempo de resolução é de 1,4 segundos.

O cenário 8 possui as mesmas restrições do cenário 1 com a área máxima de corte igual a 20 hectares, o número de restrições de blocos de colheita aumenta com relação ao cenário 7, uma vez que existe a possibilidade de colheita de dois talhões adjacentes no mesmo ano, o tempo de resolução é de 1,3 segundos e o VPL é 1% menor que o cenário 1.

No cenário 9 a área máxima foi aumentada para 30 hectares, relaxando mais o problema, mas conseqüentemente aumentando o número de restrições, pois agora existem mais combinações de blocos de colheita. O VPL aumenta, pois o problema fica menos “exigente” com a dispersão da colheita.

Os cenários 10 e 11 seguem a mesma tendência, com a área máxima cada vez menos exigente, o valor do VPL fica mais próximo do cenário 1, até o cenário 12, com 65 hectares de área máxima, apresentar VPL igual ao resultado do cenário 1. Isso acontece pois a solução do cenário 1 não possui área contínua maior que 65 hectares, portanto nenhuma das restrições de adjacência de área máxima contínua chegou a restringir a resposta. O número de restrições aumenta conforme a área máxima contínua cresce, pois existem mais possibilidades de combinações de unidades de manejo.

O cenário 13 tem as restrições de área mínima e máxima ativas ao mesmo tempo, sendo assim o cenário mais complexo a ser resolvido. O VPL é 2,5% menor que o do cenário 1. Uma compilação dos resultados é apresentada no Quadro e na FIGURA 7.

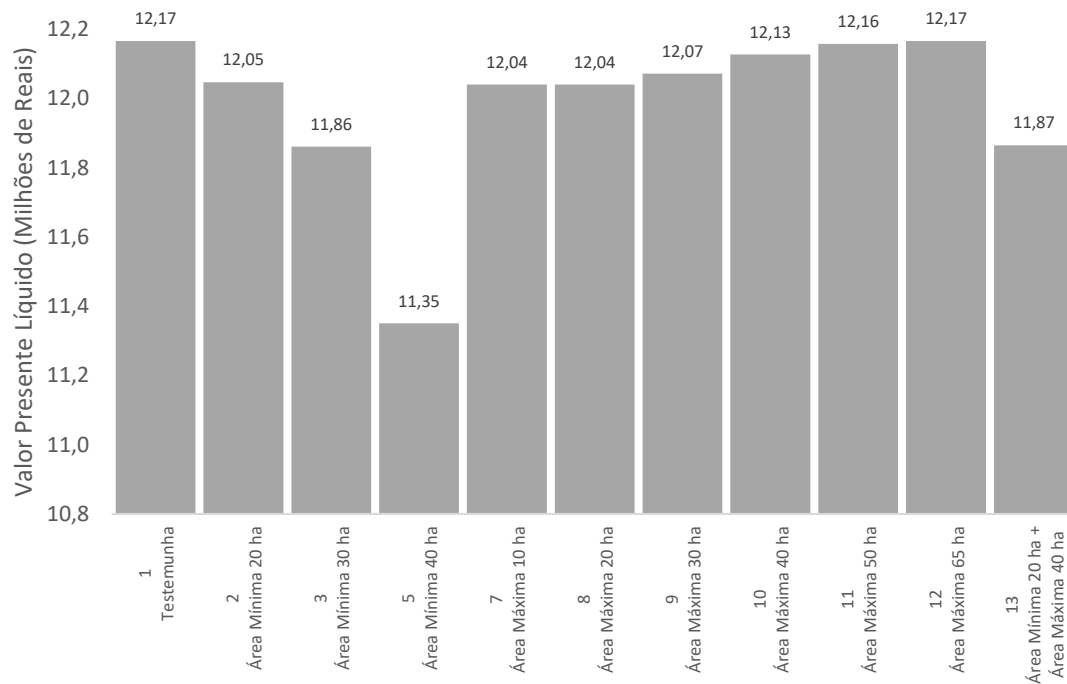


FIGURA 7 - COMPARAÇÃO DO VPL DOS CENÁRIOS

Fonte: o autor (2015)

QUADRO 4 - RESUMO DOS RESULTADOS DOS CENÁRIOS TESTADOS

Cenário	Equilíbrio (%)	Demanda (mil m ³)	Idade mínima (anos)	Área Mínima (ha)	Área Máxima (ha)	Valor presente líquido (milhares R\$)	Percentual em comparação com o cenário 1	Acréscimo no tempo de resolução em comparação com o cenário 1	Número de variáveis ativas	Número de restrições
1	±10	20	5	-	-	12.166	100,0%	1,0 x	126	42
2	±10	20	5	20	-	12.047	99,0%	19,5 x	126	136
3	±10	20	5	30	-	11.861	97,5%	94,7 x	126	168
4	±10	20	5	40	-	Infactível	-	-	117	159
5	±10	20	4	40	-	11.351	93,3%	31,6 x	137	179
6	±50	-	-	50	-	Infactível	-	-	156	192
7	±10	20	5	-	10	12.040	99,0%	3,5 x	126	292
8	±10	20	5	-	20	12.040	99,0%	3,1 x	126	291
9	±10	20	5	-	30	12.072	99,2%	9,7 x	126	383
10	±10	20	5	-	40	12.127	99,7%	44,1 x	126	742
11	±10	20	5	-	50	12.158	99,9%	56,4 x	126	1385
12	±10	20	5	-	65	12.166	100,0%	272,1 x	126	3661
13	±10	20	5	20	40	11.865	97,5%	325,9 x	126	836

Fonte: O autor (2015)

4.4. CONCLUSÕES

O modelo proposto pode ser resolvido em poucos segundos usando os recursos computacionais mencionados para todos os cenários apresentados.

O Valor Presente Líquido da Floresta decresce quando a área mínima contínua exigida pelo modelo aumenta. Exigir que blocos com áreas mínimas maiores sejam criados faz com que as unidades de manejo deixem de ser colhidas quando seu retorno econômico é máximo, consequentemente piorando o valor da função objetivo, porém os custos com deslocamento da frente de colheita e os custos administrativos de gerenciamento da colheita tendem a diminuir quando são formados blocos de colheita respeitando as áreas operacionais mínimas.

O Valor Presente Líquido da Floresta aumenta quando a área máxima contínua exigida pelo modelo aumenta. Ao aumentar o limite de área máxima o modelo fica mais 'relaxado' e consequentemente a liberdade de escolha de áreas de corte aumenta até o ponto onde as unidades de manejo escolhidas para corte são as mesmas do cenário sem restrições, portanto, se a área máxima de colheita for muito grande (maior que o maior bloco de talhões criado pelo modelo sem restrições de área máxima) as restrições de área máxima contínua não irão afetar o valor da função objetivo.

Os tempos de resolução pioram com o aumento da área mínima ou da área máxima contínua. Para a área mínima o tempo de resolução do problema piora devido à dificuldade em criar os blocos de colheita, enquanto que quando a área máxima aumenta o número de restrições também cresce, causando o decréscimo no tempo de resolução do problema.

5. APLICAÇÃO DO MODELO DE OTIMIZAÇÃO FLORESTAL ESPACIAL EM UM POVOAMENTO COMERCIAL

5.1. OBJETIVOS

- Aplicar o modelo apresentado no capítulo 4 em um plantio comercial;
- Avaliar o impacto na redução do Valor Presente Líquido que a floresta apresenta sob várias condições de restrições de área máxima e/ou mínima, assim como o impacto no tempo necessário para concluir a busca pela solução ótima do modelo;

5.2. MATERIAL E MÉTODOS

5.2.1. Área de Estudo

Os dados são provenientes de um povoamento comercial de *Pinus taeda* com diversas idades situado no estado do Paraná, Brasil, pertencente à empresa Arauco Forest Brasil, atualmente manejado com a finalidade de produção de matéria prima para painéis de madeira (MDF e MDP) e toras para serraria, com rotação desejada em torno dos 15 anos.

As áreas utilizadas nesta seção são referentes à regional de Campo do Tenente, representadas na FIGURA 8. A área produtiva de Campo do Tenente é de cerca de 10 mil hectares, distribuída em diversas fazendas.

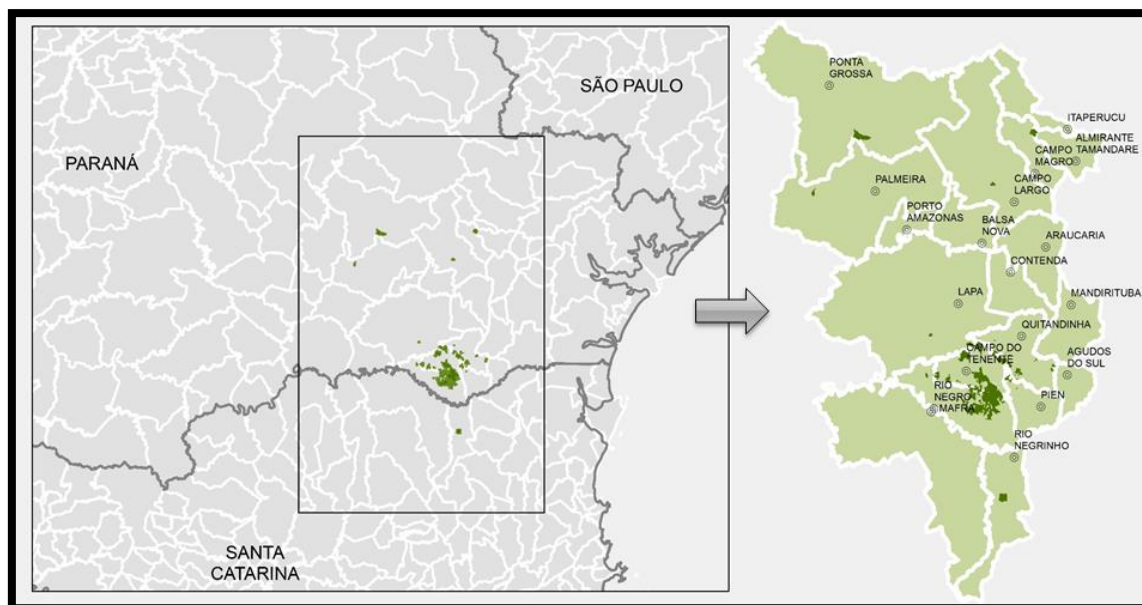


FIGURA 8 - LOCALIZAÇÃO DA ÁREA DE ESTUDO.

Fonte: ARAUCO (2015)

As áreas da região de Campo do Tenente estão inseridas no primeiro e segundo planalto Paranaense, na porção Sul-Sudeste do estado. Os solos predominantes são de baixa fertilidade natural, baixa profundidade, com predominância de Cambissolos associados com Argissolos e Latossolos. É comum encontrar afloramentos rochosos nesta região, tanto de rocha sã quanto de rocha com início de intemperização. O clima, de acordo com a classificação de Köppen é Cfb, ou seja, clima temperado, temperatura média do mês mais frio abaixo de 18 °C (mesotérmico), com geadas, verões frescos, temperatura média do mês mais quente abaixo de 22 °C e sem estação seca definida (ARAUCO, 2015).

O povoamento já possui um planejamento estratégico de longo prazo estabelecido com o auxílio do programa OpTimber - LP, utilizando um modelo de manejo tipo I com horizonte de planejamento de 30 anos, função objetivo visando a maximização do VPL e restrições de idade mínima e máxima de corte, equilíbrio do fornecimento de produtos (não-declínio da oferta de madeira), entre outras. O problema linear gerado pelo OpTimber - LP é resolvido pelo programa Lingo v. 13.0.

A prognose da produção foi feita utilizando tabelas de produção fornecidas pela empresa ajustadas para as fazendas da região de Campo do Tenente.

Talhões com o mesmo ano de plantio, mesma produtividade e adjacentes foram agrupados, diminuindo assim o número de unidades de manejo a serem enviadas ao modelo. Isto reduziu o tamanho do problema de mais de 2000 talhões para cerca de 500 unidades de manejo. Unidades de manejo que ficaram com uma área muito grande (maior que 30% da área máxima contínua) foram propositalmente separadas (mantendo as características semelhantes, exceto pela localização espacial) para que a decisão de agrupar ou não áreas tão grandes seja realizada pelo otimizador.

A idade de rotação desejada (tempo entre o plantio e o corte raso) é em torno dos 15 anos e como é possível observar na FIGURA 9 o povoamento tem uma distribuição de áreas por idade com poucos 'picos' e 'vales' (se aproxima de uma distribuição constante) e classes que vão até a idade de rotação desejada (não está concentrada no início ou meio da distribuição).

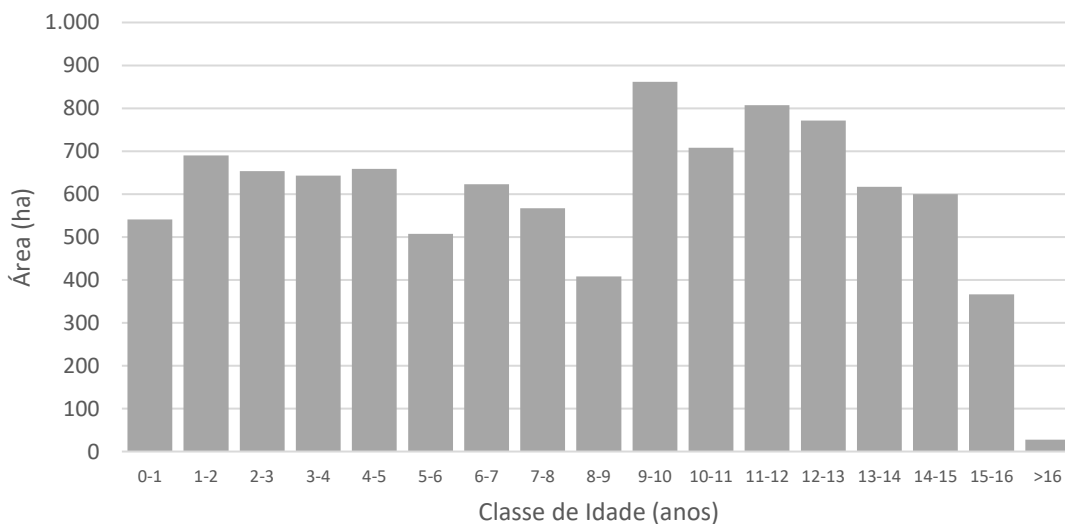


FIGURA 9 - DISTRIBUIÇÃO DAS IDADES DO POVOAMENTO

Fonte: o autor (2015)

5.2.2. Restrições de Adjacência

A adjacência entre talhões pode ter várias definições, como talhões que compartilham um ponto ou um lado de seu perímetro em comum, ou então é adotada uma distância máxima entre os centroides dos polígonos, mas a adjacência considerada neste trabalho necessita incluir mais algumas características:

- Talhões que possuem estradas entre eles devem ser considerados adjacentes, uma vez que é fácil para a máquina que realiza a colheita atravessar uma estrada.
- Talhões que possuem um rio entre eles em toda a sua extensão não devem ser considerados adjacentes, pois não é possível atravessar um rio ou córrego com as máquinas de colheita.

Desta forma foi necessária uma metodologia automatizada para encontrar uma lista de todas as unidades de manejo adjacentes e para tal foi utilizado um *buffer* de 20 metros nos polígonos que constituem as áreas produtivas. Este *buffer* é suficiente para fazer com que talhões separados por uma estrada sejam sobrepostos, mas ainda é pequeno o suficiente para manter as unidades de manejo separadas por rios sem pontos em comum, uma vez que as áreas de preservação permanente (APP) na beira dos rios têm uma largura mínima de 30 metros para cada lado da margem do rio ou córrego.

O *software* de geoprocessamento QGIS possui uma capacidade de interpretar trechos de código programável (*scripts*) em linguagem de programação *Python* (*PyQGIS*), e usando um código adaptado do *script* apresentado por Gandhi (2014) foi possível automatizar o processo de detecção de adjacências, gerando uma lista de adjacências para cada talhão.

Posteriormente à geração da lista de adjacência foi utilizada em uma planilha eletrônica e processada por uma rotina em VBA dentro do Excel para que fosse transformada em uma matriz de adjacência, formato necessário para facilitar a interpretação pela linguagem AMPL e processamento pelo algoritmo de caminhamento (*path*), adaptado de McDill et al. (2002).

5.2.3. Horizonte de Planejamento

O horizonte de planejamento utilizado foi de 6 anos, pois a resposta desejada do modelo é o agrupamento e/ou a dispersão de unidades de manejo colhidas em um mesmo ano (nível tático de planejamento). Para isso foi adicionada uma restrição de demanda mínima anual que é o limite inferior de produção praticado atualmente pela empresa e que foi estabelecido usando-se um modelo de planejamento de longo prazo.

Outro motivo para o período curto foi a necessidade de que as unidades de manejo sejam colhidas com um intervalo máximo de idade entre eles, impedindo assim a colheita de unidades de manejo com árvores de formatos muito diferentes no mesmo momento. Isto permitiu que muitas unidades de manejo fossem retiradas do modelo (aquelas que não atingiriam a idade mínima de corte dentro do Horizonte de Planejamento), facilitando assim a resolução do problema: de cerca de 500 unidades de manejo restaram apenas 254 unidades de manejo.

5.2.4. Modelo de Otimização

O conjunto de equações que forma o modelo de otimização é o mesmo apresentado no item 4.2.3, com função objetivo maximizando o VPL. O valor da demanda mínima anual foi fornecida pelo planejamento estratégico da empresa, conforme descrito no item 5.2.1.

5.2.5. Cenários

Foram criados diversos cenários para representar diferentes opções de planejamento, cada um com um diferente conjunto de restrições e/ou um valor diferente para alguma restrição. O primeiro cenário não possui restrições de adjacência ativas e foi estabelecido como o cenário “testemunha”, uma vez que por ser menos restritivo apresenta o maior Valor Presente Líquido entre os cenários. A relação dos cenários criados é apresentada no QUADRO 5.

QUADRO 5 - CENÁRIOS GERADOS PARA A FLORESTA COMERCIAL

Cenário	Equilíbrio (%)	Demanda Mínima	Idade Mínima (anos)	Área Mínima (ha)	Área Máxima (ha)
1	±10	Ativa	15	-	-
2	±10	Ativa	15	20	-
3	±10	Ativa	15	40	-
4	±10	Ativa	15	60	-
5	±10	Ativa	15	-	20
6	±10	Ativa	15	-	80
7	±10	Ativa	15	-	140
8	±10	Ativa	15	20	40

Fonte: O autor (2015)

5.2.6. Recursos Computacionais Utilizados

Os problemas foram construídos em uma planilha eletrônica com o auxílio do programa Microsoft Office Excel 2013. O modelo matemático foi escrito em linguagem AMPL utilizando-se para tal o suplemento do Excel chamado SolverStudio v0.09.01 (MASON, 2013). O otimizador utilizado foi o CBC v2.9.4 (*Coin-or branch and cut*), desenvolvido pela COIN-OR (*Computational Infrastructure for Operations Research*).

O hardware utilizado para a resolução foi um dos servidores NEOS, localizado na Universidade de Wisconsin: dois processadores Intel Xeon X5660 @ 2.8GHz, com 64 GB de memória RAM. Versão completa da AMPL instalada e restrito a problemas com até 3GB de tamanho ou oito horas de resolução.

5.3. RESULTADOS

As 254 unidades de manejo definidas no item 5.2 apresentaram poucas adjacências, como pode ser observado nas figuras FIGURA 10 e FIGURA 11, as quais mostram que muitas unidades não possuem unidades de manejo vizinhas, implicando na impossibilidade de formar blocos tanto de área máxima quanto de área mínima. 34% da área disponível para corte não é influenciada pelas restrições por não apresentar nenhuma adjacência, este número equivale a 140 das 254 unidades de manejo.

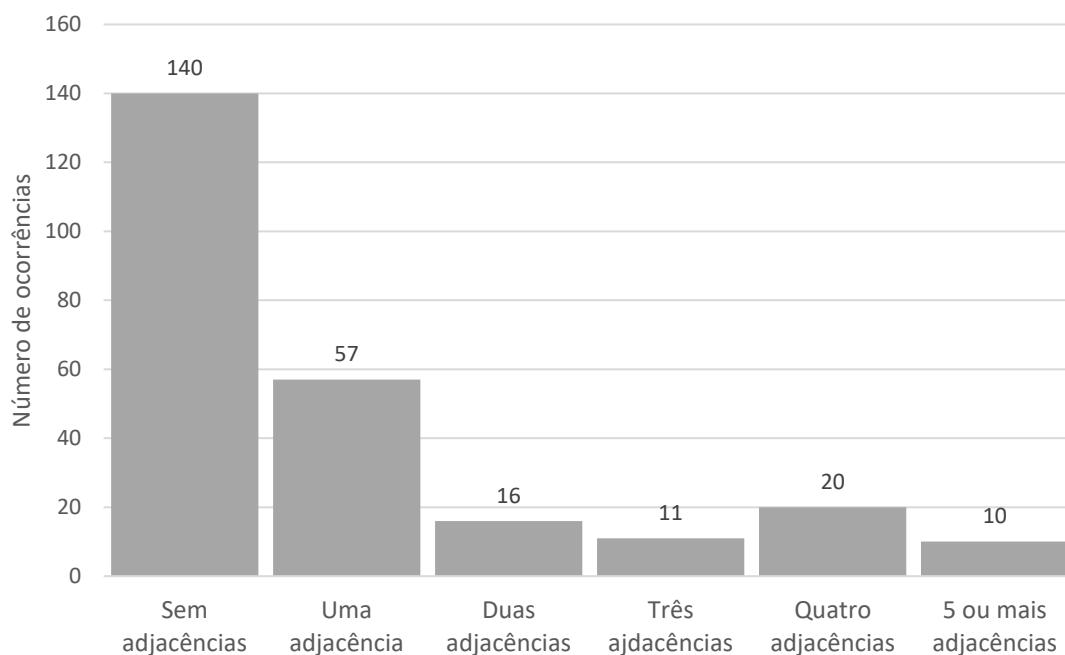


FIGURA 10 - NÚMERO DE UNIDADES DE MANEJO SEGUNDO A QUANTIDADE DE ADJACÊNCIAS.

Fonte: o autor (2015)

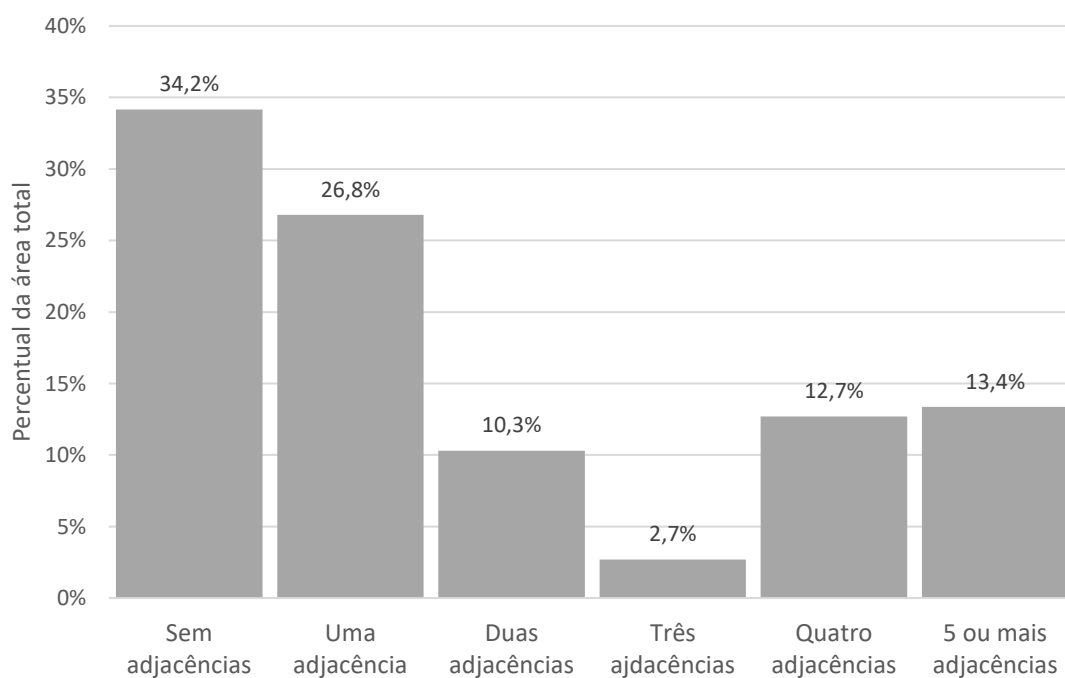


FIGURA 11 - PERCENTUAL DA ÁREA TOTAL SEGUNDO A QUANTIDADE DE ADJACÊNCIAS.

Fonte: o autor (2015)

O cenário 1 apresenta um tempo de resolução de menos de um segundo e consegue atender à todas as restrições. O Valor Presente Líquido obtido neste

cenário é considerado como sendo 100% para comparação com os demais cenários.

O cenário 2 com 20 hectares de área mínima contínua apresenta uma redução de 1,3% no valor da função objetivo. Houve um aumento no número de restrições do cenário, o que ocasiona a impossibilidade de algumas variáveis assumirem o valor 1, que neste caso diminuiu o espaço de busca pelo ótimo e reduziu o tempo necessário para o otimizador finalizar a busca.

Os cenários 3 e 4 possuem um número menor de variáveis e restrições que o cenário 2, isto se deve ao fato de que esta formulação cria apenas uma restrição de área mínima para cada unidade de manejo (equação 5.7 do item 5.2.4.2.5). Ao aumentar a área mínima algumas unidades de manejo não podem mais ser cortadas sozinhas em certos períodos, excluindo do espaço de busca as variáveis que representam o corte daquela unidade no respectivo ano do horizonte de planejamento. Conseqüentemente algumas restrições relacionadas exclusivamente àquela variável também são removidas. Entretanto essa diminuição no espaço de busca não é suficiente para diminuir o tempo de execução do otimizador, pois a restrição que faz com que a área mínima seja de 40 ou 60 hectares torna a busca pelo ótimo mais complexa, aumentando o tempo para encontrar a resposta ótima. O valor da função objetivo é 3,5% menor para o cenário 3 e 6,2% menor para o cenário 4.

O cenário 5 apresentou uma redução de 12,3% no valor da função objetivo em comparação com o cenário 1. O número de variáveis não decresce com a inclusão de restrições de área máxima, mas o número de restrições mais do que dobra em comparação com o cenário 1. O tempo de execução foi maior, mas não chega a ser considerado um tempo impeditivo operacionalmente. Os cenários 6 e 7 apresentaram comportamento semelhante, demonstrando que ao problema ficar mais 'relaxado' com a restrição de área máxima de colheita o tempo de execução volta a cair, mesmo após o acréscimo de mais restrições. O número de restrições nestes casos é elevado devido a necessidade de contemplar todos os blocos de unidades de manejo que ultrapassam a área máxima permitida, pois para cada bloco é necessária uma restrição; por outro lado, o espaço de busca aumenta, pois existem mais combinações possíveis de unidades de manejo que formam blocos de colheita factíveis, aumentando o

número de possibilidades que podem inclusive aumentar o valor da função objetivo.

O cenário 8 combina as restrições de área máxima e mínima contínuas, consequentemente este é o problema que leva mais tempo para ser resolvido, mais de 100 vezes o tempo do cenário 1, porém ainda considerado aceitável do ponto de vista operacional do modelo. O valor presente líquido apresentou uma queda de 14,6% em comparação com o cenário 1.

Todos os cenários foram resolvidos em tempo inferior a dez minutos, portanto o tempo de resolução não foi um fator restritivo para os problemas considerados.

O Quadro 5 apresenta uma compilação dos resultados e a FIGURA 12 apresenta um comparativo gráfico dos valores obtidos pela função objetivo em cada cenário.

QUADRO 6 - RESUMO DOS RESULTADOS DOS CENÁRIOS TESTADOS

Cenário	Equilíbrio (%)	Demanda Mínima	Idade Mínima (anos)	Área Mínima contínua (ha)	Área Máxima contínua (ha)	Percentual do VPL em comparação com o cenário 1 (%)	Acréscimo no tempo de resolução ¹¹	Número de variáveis ativas	Número de restrições
1	±10	Ativa	15	-	-	100,0%	1,0 x	729	178
2	±10	Ativa	15	20	-	98,7%	0,9 x	676	234
3	±10	Ativa	15	40	-	96,5%	1,4 x	659	231
4	±10	Ativa	15	60	-	93,8%	2,0 x	656	231
5	±10	Ativa	15	-	20	87,7%	11,7 x	729	409
6	±10	Ativa	15	-	80	87,7%	2,2 x	729	407
7	±10	Ativa	15	-	140	88,8%	2,2 x	729	711
8	±10	Ativa	15	20	40	85,4%	127 x	676	461

Fonte: O autor (2015)

¹¹ Em comparação ao cenário 1 (sem restrições de área mínima ou máxima contínuas)

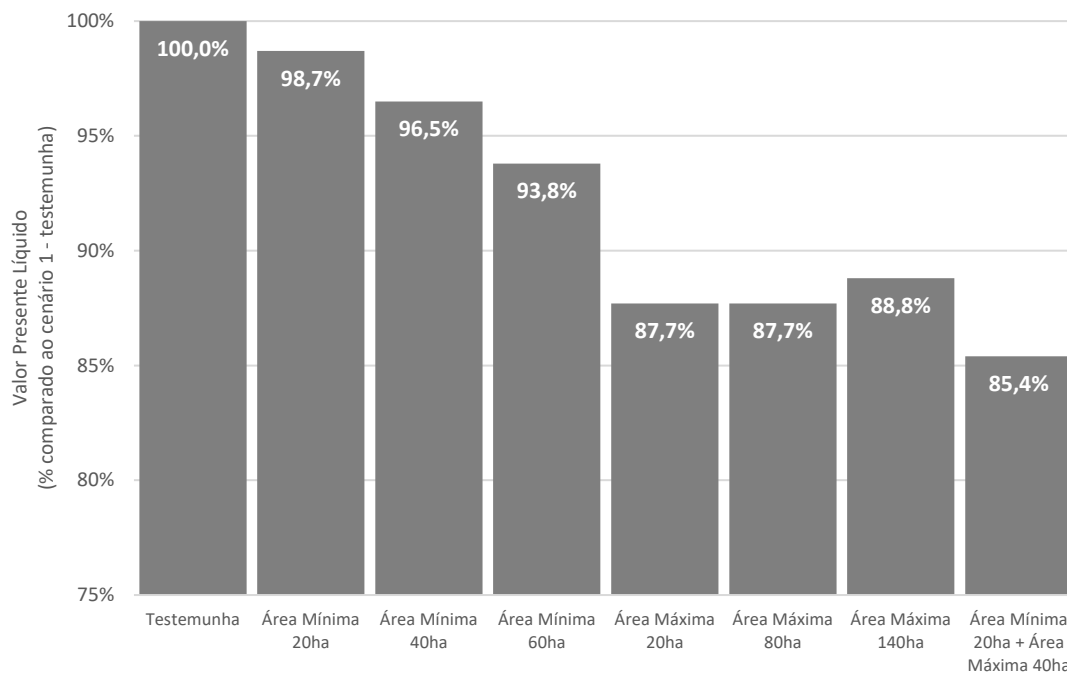


FIGURA 12 - VARIAÇÃO DO VALOR PRESENTE LÍQUIDO NOS DIFERENTES CENÁRIOS.

Fonte: o autor (2015)

Gomide (2009) utilizou restrições de área máxima contínua do tipo ARM com limitações de 50 e 70 hectares, com respectivas reduções de 2,24% e 2,20% no Valor Presente Líquido quando comparadas com o valor do cenário sem restrições espaciais.

Augustynczyk (2014) testou restrições de área máxima contínua do tipo ARM variando entre 40 e 70 hectares, com e sem equilíbrio da produção. Para os cenários com equilíbrio da produção ativo houveram resultados que tiveram um decréscimo de 1,87% a 0,01%.

Zhu et al. (2007) obtiveram decréscimos de 1% a 3,5% no valor presente líquido ao aplicar restrições de área máxima contínua em florestas de 2.942 a 5.821 hectares.

Augustynczyk (2014) também utilizou restrições de área mínima operacional variando entre 30 e 80 hectares, com diminuições no Valor Presente Líquido de 0,4% a 4,6% em relação ao cenário sem restrições espaciais.

5.4. CONCLUSÕES

O comportamento do Valor Presente Líquido (VPL) para a área comercial é semelhante ao comportamento apresentado pela área artificial apresentada no item 4. O VPL decresce conforme as restrições ficam mais impeditivas.

O tempo de resolução do problema foi aceitável em todos os cenários analisados, mas ao comparar com os tempos obtidos no item 4 é possível concluir que nem todas as conformações espaciais apresentam a mesma dificuldade em encontrar a resposta ótima, assim como o tamanho do problema pode não servir de indicativo para concluir se um problema será necessariamente mais demorado que outro; um indicativo desta situação é a pequena quantidade de unidades de manejo com muitas adjacências mostrada na FIGURA 10, como são poucas as adjacências, conseqüentemente são formadas poucas restrições.

O horizonte de planejamento adotado (6 anos) diminui o número possível de adjacências a serem formadas, pois unidades de manejo que não tem idade de corte neste período nem entram no modelo de decisão, diminuindo o espaço de busca pela resposta ótima. Entretanto, caso seja adotada a resposta gerada pelo modelo, recomenda-se rodar o modelo com mais anos do horizonte de planejamento ou então executar o modelo todo ano com no mínimo o mesmo horizonte, para que uma nova resposta seja gerada a cada ano ou período. Conhecer o comportamento da floresta por mais tempo garante que o equilíbrio da produção seja mantido, evitando grandes diferenças de oferta de produto.

Com o acréscimo do valor da área máxima contínua observou-se um aumento do VPL (em direção ao valor do cenário sem restrições de área máxima), porém há um aumento no número de restrições, pois existem mais combinações de agrupamento de unidades de manejo. Com o aumento da área mínima contínua observou-se uma diminuição no VPL e um aumento no tempo de resolução, uma vez que são necessários agrupamentos com mais unidades de manejo para atender à restrição de área mínima.

REFERÊNCIAS

- ALONSO, L. R. L. **O Problema da consideração de restrições de adjacência em um planejamento florestal**. Dissertação de Mestrado, Universidade Federal do Paraná, 2003.
- ARAUCO Forest Brasil. **Resumo público de manejo florestal: Regiões de Campo do Tenente e Sengés**. Curitiba, 2015. Acesso em fevereiro de 2015. Disponível em: http://www.arauco.cl/file/file_3305_15810-resumo-publico_2014_ct_sen_baixa_aprovado_22-12-2014.pdf.
- AUGUSTYNCZIK, A. L. D. **Planejamento Florestal Otimizado considerando áreas mínimas e máximas operacionais de colheita**. Dissertação de mestrado, Universidade Federal do Paraná, 2014.
- BETTINGER, P.; BOSTON, K.; SIRY, J. P.; GREBNER, D.L. **Forest Management and Planning**. Academic Press, 2009, 342p.
- BUONGIORNO, J.; GILLESS, K.J. **Decision methods for forest resource management**. Academic Press, 2003. 439p.
- CARVAJAL, R.; CONSTANTINO, M.; GOYCOOLEA, M.; VIELMA, J. P.; WEINTRAUB, A. **Imposing Connectivity Constraints in Forest Planning Models**. Acesso em outubro de 2014. Disponível em: http://mgoycool.uai.cl/papers/11carvajal_submitted.pdf.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to algorithms**, 3rd edition. MIT Press, Massachusetts, 2009, 1292p.
- CZYZYK, J., MESNIER, M. P., AND MORÉ, J. J. 1998. **The NEOS Server**. IEEE Journal on Computational Science and Engineering 5(3), 68-75.
- DOLAN, E. 2001. **The NEOS Server 4.0 Administrative Guide**. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory.
- FOURER, R; KERNIGHAN B. W. (2002). **AMPL: A Modeling Language for Mathematical Programming**. Duxbury Press. ISBN 978-0-534-38809-6.
- GOMIDE, L. R. **Planejamento Florestal espacial. Tese de Doutorado**, Universidade Federal do Paraná, 2009.
- GOMIDE, L. R.; ARCE, J. E.; LINDBECK, A. **Efeito das restrições de adjacência no planejamento florestal otimizado**. FLORESTA, Curitiba, PR, v. 40, n. 3, p. 573-584, jul./set. 2010.
- GOYCOOLEA, M.; MURRAY, A.; VIELMA, J. P.; WEINTRAUB, A. S. **Evaluating Approaches for Solving the Area Restriction Model in Harvest Scheduling**. Forest Science, Volume 55, Número 2, Abril 2009, pp. 149-165(17).
- GOYCOOLEA, M.; MURRAY, A.; VIELMA, J. P.; WEINTRAUB, A. S. **Evaluating Approaches for Solving the Area Restriction Model in Harvest Scheduling**. Forest Science, Volume 55, Número 2, Abril 2009, pp. 149-165(17).

GROPP, W. e MORÉ, J. J. 1997. **Optimization Environments and the NEOS Server**. Approximation Theory and Optimization, M. D. Buhmann and A. Iserles, eds., Cambridge University Press, pages 167-182

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. Porto Alegre. AMGH, 2013. 9ª edição, 1005p.

MASON, A. J. **SolverStudio: A new tool for better optimisation and simulation modelling in Excel**. INFORMS 2013. Trans. Ed. 14(1):45–52.

MCDILL, M.; REBAIN, S.; BRAZE, J. **Harvesting scheduling with area based adjacency constraints**. Forest Science 48, 2002.

MITCHELL, S. A. **Operational Forest Harvest Scheduling Optimisation – A mathematical model and solution strategy**. Auckland. 2004. 252f. Tese, University of Auckland.

NASCIMENTO, F. A. F. do. **Planejamento florestal com restrições de adjacência utilizando programação paralela** Curitiba. 2014. 130f. Tese. Universidade Federal do Paraná.

SILVA, R. T. **Planejamento florestal, modelos de programação inteira, multiobjectivo e aplicações**. Coimbra. 2004. 120f. Dissertação de Mestrado, Universidade de Coimbra.

STEIGUER, J. E. LIBERTI, L.; SCHULER, A. HANSEN, B. **Multicriteria decision models for forestry and natural resources management: an annotated bibliography. General Technical Report**. NE-307. Newtown Square, PA: U.S. Department of Agriculture, Forest Service, Northeastern Research Station. P.32. 2003.

WEINTRAUB, A.; MURRAY, A. T. **Review of combinatorial problems induced by spatial forest harvesting planning. Discrete Applied Mathematics**, vol. 154, no. 5, pp. 867–879, 2006.

ZHU, j. BETTINGER, P.; LI, R. **Additional Insight into the Performance of a New Heuristic for Solving Spatially Constrained Forest Planning Problems**. Silva Fennica 41(4): 687–698, 2007.

ANEXOS

VERSÃO DO SISEUCALIPTO UTILIZADO



FIND NEIGHBOR POLYGONS IN A LAYER (SCRIPT POR UJAVAL GANDHI, 2014)

Acessado em 17 de janeiro de 2015, disponível em http://www.qgistutorials.com/en/docs/find_neighbor_polygons.html .

```
#####
# Copyright 2014 Ujaval Gandhi
#
#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#
#This program is distributed in the hope that it will be useful,
#but WITHOUT ANY WARRANTY; without even the implied warranty of
#MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#GNU General Public License for more details.
#
#You should have received a copy of the GNU General Public
#License along with this program; if not, write to the Free
#SoftwareFoundation, Inc., 51 Franklin Street, Fifth Floor,
#Boston, MA 02110-1301, USA.
#
#####
from qgis.utils import iface
from PyQt4.QtCore import QVariant

# Replace the values below with values from your layer.
# For example, if your identifier field is called 'XYZ', then
# change the line below to _NAME_FIELD = 'XYZ'
_NAME_FIELD = 'NAME'
# Replace the value below with the field name that you want to
# sum up.
# For example, if the # field that you want to sum up is called
# 'VALUES', then change the line below to _SUM_FIELD = 'VALUES'
_SUM_FIELD = 'POP_EST'

# Names of the new fields to be added to the layer
_NEW_NEIGHBORS_FIELD = 'NEIGHBORS'
_NEW_SUM_FIELD = 'SUM'

layer = iface.activeLayer()

# Create 2 new fields in the layer that will hold the list of
# neighbors and sum of the chosen field.
layer.startEditing()
layer.dataProvider().addAttributes(
    [QgsField(_NEW_NEIGHBORS_FIELD, QVariant.String),
     QgsField(_NEW_SUM_FIELD, QVariant.Int)])
layer.updateFields()
# Create a dictionary of all features
```

```

feature_dict = {f.id(): f for f in layer.getFeatures()}

# Build a spatial index
index = QgsSpatialIndex()
for f in feature_dict.values():
    index.insertFeature(f)

# Loop through all features and find features that touch each
# feature
for f in feature_dict.values():
    print 'Working on %s' % f[_NAME_FIELD]
    geom = f.geometry()
    # Find all features that intersect the bounding box of the
    # current feature.
    # We use spatial index to find the features intersecting the
    # bounding box of the current feature. This will narrow down
    # the features that we need to check neighboring features.
    intersecting_ids = index.intersects(geom.boundingBox())
    # Initialize neighbors list and sum
    neighbors = []
    neighbors_sum = 0
    for intersecting_id in intersecting_ids:
        # Look up the feature from the dictionary
        intersecting_f = feature_dict[intersecting_id]

        # For our purpose we consider a feature as 'neighbor' if
        # it touches or intersects a feature. We use the
        # 'disjoint' predicate to satisfy these conditions. So
        # if a feature is not disjoint, it is a neighbor.
        if (f != intersecting_f and
            not intersecting_f.geometry().disjoint(geom)):
            neighbors.append(intersecting_f[_NAME_FIELD])
            neighbors_sum += intersecting_f[_SUM_FIELD]
    f[_NEW_NEIGHBORS_FIELD] = ','.join(neighbors)
    f[_NEW_SUM_FIELD] = neighbors_sum
    # Update the layer with new attribute values.
    layer.updateFeature(f)

layer.commitChanges()
print 'Processing complete.'

```

MODELO DE MANEJO EM AMPL

```
##### Modelo de Manejo
### Definicoes dos parametros e variaveis principais

# Vetor dos anos do horizonte de planejamento (HP)
set AnosHP ordered;

# Vetor com as areas minimas de corte para cada ano do HP
param AreasMinCorteAnual{AnosHP}>=0;

# Vetor com as areas maximas de corte para cada ano do HP
param AreasMaxCorteAnual{AnosHP}>=0;

# Vetor com as demandas minimas de madeira para cada ano do HP
param DemandasMin{AnosHP}>=0;

# Vetor com as ofertas maximas de madeira para cada ano do HP
param OfertasMax{AnosHP}>=0;

# Vetor com os Talhoes
set Talhoes ordered;

# Celula com a Taxa de Juros a ser aplicada
param TaxaJuros>0;

# Celula que guarda a idade minima de corte
param IdadeMinimaCorte>=0;

# Celula com o preco atual da madeira
param PrecoMadeira>0;

# Vetor com as areas de cada talhao
param AreasTalhoes{Talhoes}>=0;

# Vetor com as idades das unidades de manejo
param Idades{Talhoes}>=0;

# Matriz com as produtividades esperadas em cada ano do HP
param Produtividades{Talhoes,AnosHP}>=0;

# Matriz com as identificacoes de cortes (variaveis)
var Cortes{Talhoes,AnosHP} binary;

### Definicao dos parametros necessarios para a Adjacencia
# Matriz que guarda as adjacencias dos talhoes
param MatrizAdjacencia{Talhoes, Talhoes} binary, default 0;

# Celula que define a area maxima para o bloco de colheita
param AreaMaxima >= 0;

# Criacao de um vetor para guardar as restricoes de adjacencia
set RestricoesAdjacencia{1..1000000} default {};
```

```

# Criacao de um contador para o loop do path
param n >= 0, integer, default 0;
param _n >= 0, integer, default 0;
param CaminhoRepetido >= 0, integer, default 0;

# Criacao de um vetor que guardara o caminho percorrido pelo
path
set Caminho default {};

# Criacao de uma lista de adjacencia para cada talhao (a lista e
mais eficiente que a matriz de adjacencia)
set ListaAdjacencia{Talhoes} default {};

# Criacao de uma lista com uma area maxima de adjacencia para o
bloco do Talhao i
param ListaAreasMaximasAdjacentes{Talhoes} default 0;

# Celula que guarda a area minima do bloco
param AreaMinima >= 0;

# Trazer os valores da planilha
data SheetData.dat;

### Crie uma lista de adjacencia
# Para cada talhao existente
for {i in Talhoes} {
  # Adicione ele proprio a lista
  let ListaAdjacencia[i] := {i};
  # Para cada um dos demais talhoes
  for {ii in Talhoes} {
    # Se o talhao ii for adjacente ao Talhao i
    if MatrizAdjacencia[i,ii] = 1 then {
      # Adicione o talhao ii a lista de adjacencia do talhao i
      let ListaAdjacencia[i] := ListaAdjacencia[i] union {ii};
    }
  }
}

### Criar uma lista com as areas maximas dos blocos por talhao
# Para cada talhao existente
for {i in Talhoes} {
  # Faca com que a soma dos talhoes adjacentes ao talhao i
corresponda a area maxima do bloco a ser formado para o talhao i
  let ListaAreasMaximasAdjacentes[i] := sum{ii in
ListaAdjacencia[i]} AreasTalhoes[ii];
}

### Path algorithm para restricoes de area maxima
# faca com que o contador inicie no 1
let n := 1;

# Para cada talhao existente
for {i1 in Talhoes} {
  # Adicione o talhao i ao caminho
  let Caminho := {i1};
  # para cada talhao adjacente ao talhao i1

```

```

for {i2 in ListaAdjacencia[i1]} {
  # se o talhao ainda nao estiver no caminho
  if not (i2 in Caminho) then {
    # adicione o talhao ao caminho
    let Caminho := Caminho union {i2};
    # se a soma das areas dos talhoes no caminho for maior ou
igual a area maxima
    if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then {
      # entao escreva uma restricao para esse caminho
      let _n := 1;
      let CaminhoRepetido := 0;
      repeat {
        let _n := _n + 1;
        if _n >= n then break;
        if RestricoesAdjacencia[_n] within Caminho then {
          let CaminhoRepetido := 1;
          break;
        }
      };
      if CaminhoRepetido = 0 then {
        let RestricoesAdjacencia[n] := Caminho;
        let n := n + 1;
      }
    } else {
      # caso a soma das areas dos talhoes no caminho for menor que
a area maxima
      # para cada talhao adjacente a ambos os talhoes i1 e i2
      for {i3 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]}}
      {
        # repita o processo ...
        if not (i3 in Caminho) then {
          let Caminho := Caminho union {i3};
          if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
          {
            let _n := 1;
            let CaminhoRepetido := 0;
            repeat {
              let _n := _n + 1;
              if _n >= n then break;
              if RestricoesAdjacencia[_n] within Caminho then {
                let CaminhoRepetido := 1;
                break;
              }
            };
            if CaminhoRepetido = 0 then {
              let RestricoesAdjacencia[n] := Caminho;
              let n := n + 1;
            }
          }
        } else {
          for {i4 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3]}} {
            if not (i4 in Caminho) then {
              let Caminho := Caminho union {i4};
              if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
              {
                let _n := 1;

```

```

let CaminhoRepetido := 0;
repeat {
  let _n := _n + 1;
  if _n >= n then break;
  if RestricoesAdjacencia[_n] within Caminho then {
    let CaminhoRepetido := 1;
    break;
  }
};
if CaminhoRepetido = 0 then {
  let RestricoesAdjacencia[n] := Caminho;
  let n := n + 1;
}
} else {
  for {i5 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] }} {
    if not (i5 in Caminho) then {
      let Caminho := Caminho union {i5};
      if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
{
        let _n := 1;
        let CaminhoRepetido := 0;
        repeat {
          let _n := _n + 1;
          if _n >= n then break;
          if RestricoesAdjacencia[_n] within Caminho then {
            let CaminhoRepetido := 1;
            break;
          }
        };
        if CaminhoRepetido = 0 then {
          let RestricoesAdjacencia[n] := Caminho;
          let n := n + 1;
        }
      } else {
        for {i6 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] union
ListaAdjacencia[i5]}} {
          if not (i6 in Caminho) then {
            let Caminho := Caminho union {i6};
            if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
{
              let _n := 1;
              let CaminhoRepetido := 0;
              repeat {
                let _n := _n + 1;
                if _n >= n then break;
                if RestricoesAdjacencia[_n] within Caminho then {
                  let CaminhoRepetido := 1;
                  break;
                }
              };
              if CaminhoRepetido = 0 then {
                let RestricoesAdjacencia[n] := Caminho;
                let n := n + 1;
              }
            }
          }
        }
      }
    }
  }
}

```

```

    } else {
      for {i7 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] union
ListaAdjacencia[i5] union ListaAdjacencia[i6]}} {
        if not (i7 in Caminho) then {
          let Caminho := Caminho union {i7};
          if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
{
            let _n := 1;
            let CaminhoRepetido := 0;
            repeat {
              let _n := _n + 1;
              if _n >= n then break;
              if RestricoesAdjacencia[_n] within Caminho then {
                let CaminhoRepetido := 1;
                break;
              }
            };
            if CaminhoRepetido = 0 then {
              let RestricoesAdjacencia[n] := Caminho;
              let n := n + 1;
            }
          } else {
            for {i8 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] union
ListaAdjacencia[i5] union ListaAdjacencia[i6] union
ListaAdjacencia[i7]}} {
              if not (i8 in Caminho) then {
                let Caminho := Caminho union {i8};
                if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
{
                  let _n := 1;
                  let CaminhoRepetido := 0;
                  repeat {
                    let _n := _n + 1;
                    if _n >= n then break;
                    if RestricoesAdjacencia[_n] within Caminho then {
                      let CaminhoRepetido := 1;
                      break;
                    }
                  };
                  if CaminhoRepetido = 0 then {
                    let RestricoesAdjacencia[n] := Caminho;
                    let n := n + 1;
                  }
                } else {
                  for {i9 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] union
ListaAdjacencia[i5] union ListaAdjacencia[i6] union
ListaAdjacencia[i7] union ListaAdjacencia[i8]}} {
                    if not (i9 in Caminho) then {
                      let Caminho := Caminho union {i9};
                      if sum{_i in Caminho} AreasTalhoes[_i] >= AreaMaxima then
{
                        let _n := 1;
                        let CaminhoRepetido := 0;

```

```

repeat {
  let _n := _n + 1;
  if _n >= n then break;
  if RestricoesAdjacencia[_n] within Caminho then {
    let CaminhoRepetido := 1;
    break;
  }
};
if CaminhoRepetido = 0 then {
  let RestricoesAdjacencia[n] := Caminho;
  let n := n + 1;
}
} else {
  for {i10 in {ListaAdjacencia[i1] union ListaAdjacencia[i2]
union ListaAdjacencia[i3] union ListaAdjacencia[i4] union
ListaAdjacencia[i5] union ListaAdjacencia[i6] union
ListaAdjacencia[i7] union ListaAdjacencia[i8] union
ListaAdjacencia[i9]}} {
    if not (i10 in Caminho) then {
      let _n := 1;
      let CaminhoRepetido := 0;
      repeat {
        let _n := _n + 1;
        if _n >= n then break;
        if RestricoesAdjacencia[_n] within Caminho union {i10}
then {
          let CaminhoRepetido := 1;
          break;
        }
      };
      if CaminhoRepetido = 0 then {
        let RestricoesAdjacencia[n] := Caminho union {i10};
        let n := n + 1;
      }
    }
  }
}
}
let Caminho := Caminho diff {i9};
}
}
let Caminho := Caminho diff {i8};
}
}
let Caminho := Caminho diff {i7};
}
}
let Caminho := Caminho diff {i6};
}
}
let Caminho := Caminho diff {i5};
}
}

```

```

    }
    let Caminho := Caminho diff {i4};
  }
}
}
let Caminho := Caminho diff {i3};
}
}
}
let Caminho := Caminho diff {i2};
}
}
}
#display n;

### Funcao Objetivo
# Maximizar o Valor Presente Liquido da Floresta
maximize VPL:
  sum {i in Talhoes, j in AnosHP} Cortes[i,j] *
  AreasTalhoes[i] * Produtividades[i,j] *
  (PrecoMadeira/((1+TaxaJuros)^j));
expand VPL;

### Restricoes
# Nao corte mais que uma vez o talhao dentro do HP
subject to AreaMaximaPorTalhao {i in Talhoes}:
  sum{j in AnosHP} Cortes[i,j] <= 1;
expand AreaMaximaPorTalhao;

# Atenda as demandas minimas para cada ano (ignore demandas
iguais a zero)
subject to DemandaMinima {j in AnosHP: DemandasMin[j] > 0}:
  sum {i in Talhoes} Cortes[i,j] * AreasTalhoes[i] *
  Produtividades[i,j] >= DemandasMin[j];
expand DemandaMinima;

# Nao oferte mais do que o maximo aceitavel para cada ano
(ignore ofertas iguais a zero)
subject to OfertaMaxima {j in AnosHP: OfertasMax[j] > 0}:
  sum {i in Talhoes} Cortes[i,j] * AreasTalhoes[i] *
  Produtividades[i,j] <= OfertasMax[j];
expand OfertaMaxima;

# Nao corte menos que uma area especifica para cada ano (ignore
as areas iguais a zero)
subject to AreaMinimaAnual {j in AnosHP: AreasMinCorteAnual[j] >
0}:
  sum{i in Talhoes} Cortes[i,j] * AreasTalhoes[i] >=
  AreasMinCorteAnual[j];
expand AreaMinimaAnual;

# Nao corte mais que uma area especifica para cada ano (ignore
as areas iguais a zero)
subject to AreaMaximaAnual {j in AnosHP: AreasMaxCorteAnual[j] >
0}:

```

```

    sum{i in Talhoes} Cortes[i,j] * AreasTalhoes[i] <=
AreasMaxCorteAnual[j];
expand AreaMaximaAnual;

# Nao corte unidades de manejo com idade menor que ___ anos
subject to IdadesMinimasdeCorte {i in Talhoes, j in AnosHP :
IdadeMinimaCorte > 0}:
    Cortes[i,j] * (Idades[i] + j) >= Cortes[i,j] *
IdadeMinimaCorte;
expand IdadesMinimasdeCorte;

# Faca com que o volume de madeira nao decline durante o periodo
#subject to VolumeCrescente {j in AnosHP: j > 0}:
#    sum{i in Talhoes}
Cortes[i,j]*AreasTalhoes[i]*Produtividades[i,j] >= sum{i in
Talhoes} Cortes[i,j-1]*AreasTalhoes[i]*Produtividades[i,j-1];
#expand VolumeCrescente;

subject to EquilibrioVolumeP {j in AnosHP: j > 0}:
    sum{i in Talhoes}
Cortes[i,j]*AreasTalhoes[i]*Produtividades[i,j] <= sum{i in
Talhoes} Cortes[i,0]*AreasTalhoes[i]*Produtividades[i,0]*1.1;
expand EquilibrioVolumeP;

subject to EquilibrioVolumeN {j in AnosHP: j > 0}:
    sum{i in Talhoes}
Cortes[i,j]*AreasTalhoes[i]*Produtividades[i,j] >= sum{i in
Talhoes} Cortes[i,0]*AreasTalhoes[i]*Produtividades[i,0]*0.9;
expand EquilibrioVolumeN;

# Faca com que o VPL nao decline durante o periodo
#subject to VPLCrescente {j in AnosHP: j > 0}:
#    sum{i in Talhoes}
Cortes[i,j]*AreasTalhoes[i]*Produtividades[i,j]*(PrecoMadeira/((
1+TaxaJuros)^j)) >= sum{i in Talhoes} Cortes[i,j-
1]*AreasTalhoes[i]*Produtividades[i,j-
1]*(PrecoMadeira/((1+TaxaJuros)^(j-1)));
#expand VPLCrescente;

# decemente o contador em uma unidade
let n := n - 1;

### Restricao de Area Maxima para cada bloco de colheita
# O bloco da restricao nao pode possuir corte de todos os seus
talhoes em um unico ano do HP

subject to AdjacenciasAreaMaxima {p in 1..n, j in AnosHP :
AreaMaxima > 0}:
    sum{i in RestricoesAdjacencia[p]} Cortes[i,j] <=
card{RestricoesAdjacencia[p]} - 1;
if AreaMaxima > 0 then {
    for {p in 1..n} {
        expand AdjacenciasAreaMaxima[p,0];
    };
};
### Restricao de Area Minima

```

```
# A soma das areas dos talhoes adjacentes ao Talhao i em cada
ano do HP, caso ele seja cortado, deve ser maior ou igual a area
minima
subject to AdjacenciasAreaMinima {j in AnosHP, i in Talhoes :
AreaMinima > 0}:
    sum{ii in ListaAdjacencia[i]} Cortes[ii,j] *
AreasTalhoes[ii]
    >= min(AreaMinima, ListaAreasMaximasAdjacentes[i]) *
Cortes[i,j];
expand AdjacenciasAreaMinima;

#Ajustar o tempo maximo para pouco menos de 8 horas, limite do
servidor NEOS
#option cbc_options "sec=28700";
solve;
```