

LUCIENE REGINA LEINEKER ALONSO

O PROBLEMA DA CONSIDERAÇÃO DE RESTRIÇÕES DE  
ADJACÊNCIA EM UM PLANEJAMENTO FLORESTAL

CURITIBA

2003

**LUCIENE REGINA LEINEKER ALONSO**

**O PROBLEMA DA CONSIDERAÇÃO DE RESTRIÇÕES DE  
ADJACÊNCIA EM UM PLANEJAMENTO FLORESTAL**

**Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências. Programa de pós-graduação em Métodos Numéricos em Engenharia – Programação Matemática, Setores de Tecnologia e Ciências Exatas, Universidade Federal do Paraná.  
Orientador: Prof.º Dr. Celso Carnieri.**

**CURITIBA**

**2003**

LUCIENE REGINA LEINEKER ALONSO

**O PROBLEMA DA CONSIDERAÇÃO DE RESTRIÇÕES DE  
ADJACÊNCIA EM UM PLANEJAMENTO FLORESTAL**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná, pela comissão formada pelos professores:

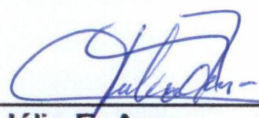
Orientador:



Prof<sup>o</sup> Dr. Celso Carnieri  
Programa de Pós-Graduação em Métodos  
Numéricos em Engenharia - UFPR



Roberto M. Profil  
Departamento de Administração – PUC-PR



Júlio E. Arce  
Departamento de Ciências Florestais - UFPR



Neida M. P. Volpi  
Departamento de Matemática - UFPR

Curitiba, 28 de maio de 2003.

**Aos meus pais, marido e filhos.**

## AGRADECIMENTOS

À Deus que me deu força para iniciar e prosseguir nesta caminhada, enfrentando e superando as dificuldades encontradas durante o caminho.

Ao professor Celso Carnieri, pelo estímulo, amizade, paciência e orientação deste trabalho.

Aos meus pais, Antonio e Maria Lúcia, que sempre me incentivaram e apoiaram.

Ao meu marido Fernando, que esteve ao meu lado apoiando e incentivando, principalmente nos momentos difíceis.

Aos meus filhos Diego e Gabriele, pela compreensão nas horas de ausência.

Aos professores do Curso de Métodos Numéricos em Engenharia, que se mostraram dispostos a sanar as dúvidas que surgiram.

Aos colegas de turma, que ajudaram a superar as dificuldades iniciais do estudo.

Ao Departamento de Matemática da Unicentro, por viabilizar a minha participação no curso e em especial ao professor Osmar que não mediu esforços para que o mesmo acontecesse.

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>vii</b>
<b>LISTA DE TABELAS .....</b>	<b>viii</b>
<b>LISTA DE QUADROS .....</b>	<b>x</b>
<b>RESUMO.....</b>	<b>xi</b>
<b>ABSTRACT.....</b>	<b>xii</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1 Objetivo do trabalho .....	3
1.2 Importância do trabalho .....	4
1.3 Limitação do trabalho .....	5
1.4 Estrutura do trabalho .....	5
<b>2. MODELOS DA PESQUISA OPERACIONAL PARA O PLANEJAMENTO FLORESTAL.....</b>	<b>7</b>
2.1 Técnicas de análise de investimentos.....	9
2.1.1 Método dos Valor Presente Líquido .....	10
2.1.2 Método do Valor Presente Líquido Anualizado .....	11
2.2 Modelos de programação Linear para Planejamento Florestal.....	12
2.3 Técnicas para a resolução de PPLi .....	17
2.3.1 O Algoritmo “ Branch and Bound” .....	17
2.3.1.1 Descrição do algoritmo “Branch and Bound” .....	19
2.3.2 O Algoritmo de Balas .....	22
2.3.2.1 Descrição do Algoritmo de Balas .....	22
<b>3. APLICAÇÕES DE TÉCNICAS DESENVOLVIDAS PARA RESOLUÇÃO DE PROBLEMAS COM RESTRIÇÕES DE ADJACÊNCIA.....</b>	<b>27</b>
3.1 Formulação de restrições de adjacência em modelos de otimização linear para planejamento tático .....	28
3.2 Algoritmo de geração de colunas para resolver problemas gerais de planejamento florestal com restrições de adjacência .....	36
<b>4. MATERIAL E MÉTODOS .....</b>	<b>42</b>
4.1 Formulação das restrições de adjacência para evitar a colheita de talhões vizinhos em um mesmo período .....	43
4.2 Formulação das restrições de adjacência para impedir a colheita de talhões vizinhos em períodos consecutivos .....	47
4.3 Formulação das restrições de adjacência para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período .....	48
4.4 Formulação do modelo completo .....	51
4.4.1 Função Objetivo .....	51

4.4.2 Restrições .....	51
4.4.2.1 Restrições de corte .....	52
4.4.2.2 Restrições de demanda .....	52
4.5 Exemplo de aplicação das restrições de adjacência .....	54
<b>5. ANÁLISE DOS RESULTADOS .....</b>	<b>65</b>
<b>6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS ..</b>	<b>73</b>
6.1 Conclusões .....	73
6.2 Sugestões para trabalhos futuros .....	74
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>76</b>
<b>ANEXO 1 – UTILIZAÇÃO DO LINGO .....</b>	<b>80</b>
<b>ANEXO 2 - DADOS UTILIZADOS PARA O PROBLEMA COM 16 TALHÕES E 10 PERÍODOS .....</b>	<b>90</b>
<b>ANEXO 3 - FORMULAÇÃO DOS PROBLEMAS DE PLANEJAMENTO FLORESTAL COM RESTRIÇÕES DE ADJACÊNCIA UTILIZANDO A SINTAXE DO LINGO.....</b>	<b>99</b>
A - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos em um mesmo período.....	99
B - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos em períodos consecutivos.....	101
C - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período.....	103
<b>ANEXO 4 – DISTRIBUIÇÃO DOS TALHÕES UTILIZADOS NA SIMULAÇÃO E A SOLUÇÃO DE CADA SIMULAÇÃO .....</b>	<b>106</b>

## LISTA DE FIGURAS

<b>FIGURA 2.1.1</b>	<b>Representação esquemática de um fluxo de caixa .....</b>	<b>10</b>
<b>FIGURA 3.1.1</b>	<b>Delineamento da área da floresta .....</b>	<b>28</b>
<b>FIGURA 3.2.1</b>	<b>Representação esquemática de zona, área, unidade .....</b>	<b>38</b>
<b>FIGURA 4.1.1</b>	<b>Distribuição dos talhões .....</b>	<b>44</b>
<b>FIGURA 4.2.1</b>	<b>Distribuição dos talhões .....</b>	<b>47</b>
<b>FIGURA 4.3.1</b>	<b>Distribuição dos talhões .....</b>	<b>49</b>
<b>FIGURA 4.5.1</b>	<b>Distribuição dos 16 talhões .....</b>	<b>55</b>
<b>FIGURA 4.5.2</b>	<b>Janela do LINGO, Solver Status .....</b>	<b>58</b>
<b>FIGURA 4.5.3</b>	<b>Solução do problema .....</b>	<b>59</b>
<b>FIGURA 5.1</b>	<b>Talhões que devem ser colhidos nos períodos 10, 20 e 30, com restrições de adjacência impostas para talhões vizinhos em um mesmo período .....</b>	<b>70</b>
<b>FIGURA 5.2</b>	<b>Talhões que devem ser colhidos nos períodos 10, 20 e 30, com restrições de adjacência impostas a talhões vizinhos e talhões distantes .....</b>	<b>71</b>
<b>FIGURA 1</b>	<b>Definição dos conjuntos, dados e formulação de um problema no LINGO .....</b>	<b>86</b>
<b>FIGURA 2</b>	<b>Janela Solver Status do LINGO .....</b>	<b>88</b>
<b>FIGURA 3</b>	<b>Solução do problema .....</b>	<b>88</b>
<b>FIGURA 4</b>	<b>Distribuição dos 50 talhões da floresta .....</b>	<b>106</b>
<b>FIGURA 5</b>	<b>Descrição dos 32 talhões para simulação .....</b>	<b>110</b>
<b>FIGURA 6</b>	<b>Distribuição dos 25 talhões da floresta .....</b>	<b>112</b>

## LISTA DE TABELAS

TABELA 4.5.1	Demanda dos produtos 1, 2 e 3 em cada período do horizonte de planejamento.....	56
TABELA 4.5.2	Solução do problema com 16 talhões e 10 períodos.....	59
TABELA 4.5.3	Solução do problema com 16 talhões e 10 períodos, com restrições de adjacência para períodos consecutivos .....	61
TABELA 4.5.4	Solução do problema com 16 talhões e 10 períodos, com restrições de adjacência para talhões vizinhos e talhões distantes.....	63
TABELA 1	Receita anual de cada talhão .....	90
TABELA 2	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 1.....	91
TABELA 3	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 2.....	91
TABELA 4	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 3.....	92
TABELA 5	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 4.....	92
TABELA 6	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 5.....	93
TABELA 7	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 6.....	93
TABELA 8	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 7.....	94
TABELA 9	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 8.....	94
TABELA 10	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 9.....	95
TABELA 11	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 10.....	95
TABELA 12	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 11.....	96
TABELA 13	Quantidade em m <sup>3</sup> de madeira disponível para laminação, serraria e processo por período no	

	<b>talhão 12.....</b>	<b>96</b>
<b>TABELA 14</b>	<b>Quantidade em m<sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 13.....</b>	<b>97</b>
<b>TABELA 15</b>	<b>Quantidade em m<sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 14.....</b>	<b>97</b>
<b>TABELA 16</b>	<b>Quantidade em m<sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 15.....</b>	<b>98</b>
<b>TABELA 17</b>	<b>Quantidade em m<sup>3</sup> de madeira disponível para laminação, serraria e processo por período no talhão 16.....</b>	<b>98</b>
<b>TABELA 18</b>	<b>Solução da simulação para 32 talhões e 15 anos, com restrições de adjacência no mesmo período .....</b>	<b>111</b>
<b>TABELA 19</b>	<b>Solução da simulação para 32 talhões e 15 anos, com restrições de adjacência em períodos consecutivos.....</b>	<b>111</b>
<b>TABELA 20</b>	<b>Solução da simulação para 32 talhões e 15 anos, com restrições de adjacência para talhões vizinhos e talhões distantes .....</b>	<b>112</b>

**LISTA DE QUADROS**

<b>QUADRO 3.1.1</b>	<b>Resultado dos testes .....</b>	<b>35</b>
<b>QUADRO 5.1</b>	<b>Resultado dos testes com 16 talhões .....</b>	<b>66</b>
<b>QUADRO 5.2</b>	<b>Resultado dos testes com 25 talhões e 15 períodos.....</b>	<b>68</b>
<b>QUADRO 5.3</b>	<b>Resultado dos testes com 32 talhões e 15 períodos.....</b>	<b>68</b>
<b>QUADRO 5.4</b>	<b>Resultado dos testes com 50 talhões e 30 períodos.....</b>	<b>69</b>

## RESUMO

O objetivo deste trabalho é abordar o problema do uso de restrições de adjacência em um modelo de planejamento florestal de longo e médio prazo. Em um sistema de planejamento florestal, é desejável que talhões adjacentes não sejam selecionados para corte em um mesmo período ou em períodos consecutivos.

Neste estudo é proposto um modelo para otimizar o corte de árvores, respeitando as restrições de adjacência. O modelo mostrou-se também adequado para evitar que talhões muito distantes um do outro, ou de difícil acesso, fossem selecionados para corte, em um mesmo período, o que aumentaria muito o custo de produção.

Técnicas de programação linear, como a Programação Linear Inteira Binária foram utilizadas para solucionar o problema. Os resultados são comparados com trabalhos semelhantes encontrados na literatura.

Para a solução do problema foi utilizado o software LINGO. Os resultados foram avaliados levando em consideração o tempo para sua resolução e também se todas as restrições foram atendidas. Vários testes foram realizados com diferentes números de talhões e horizontes de planejamento diferentes para que se pudesse avaliar a validade da formulação proposta.

**Palavras-chave:** Planejamento Florestal, Programação Linear Inteira e Restrições de adjacência.

## ABSTRACT

When planning forest management, it's desirable to avoid tree cutting in adjacent stands, in the same or consecutive harvesting periods. In this study we propose a model to optimize tree harvesting, taking in consideration adjacency constraints. The model also proved to be useful to avoid harvest spreading, i.e., to prevent the selection of stands extremely separated to be cut in a same period, which would increase harvest costs.

Mathematical programming techniques, as binary integer linear programming, were used to solve the problem. Many cases studies were analysed, varying the number of stands and planning horizon, in order to validate the proposed model. The solutions were computationally supported by the software package "LINGO" and evaluated in terms of computational timing and the ability to accomplish the restrictions. Related works in this field are also discussed.

**Key-words:** Forest planning, Linear and Integer Programming and Adjacency Constraints.

# CAPÍTULO I

## 1. INTRODUÇÃO

Nos últimos anos houve uma grande evolução da Ciência Florestal no Brasil, o que propiciou a revisão de uma série de conceitos no que concerne ao que fazer com o recurso florestal (SCOLFORO, 1993). Tendo em vista que o plantio de florestas no Brasil é uma atividade econômica importante, faz-se necessária a utilização de técnicas que ajudem na tomada de decisão. A Pesquisa Operacional vem colaborando muito no planejamento florestal, uma vez que modelos de planejamento envolvem um número muito grande de variáveis assim como várias decisões devem ser tomadas ao longo do tempo.

SANQUETTA (1996) afirma que o planejamento florestal é muito peculiar e bastante complexo. Também, BORGES (1989) diz que no âmbito florestal essa complexidade advém do longo prazo das rotações das plantações florestais, da extensão das áreas cultivadas e da suscetibilidade das florestas às mudanças no ambiente, seja a sua causa natural ou humana. Esta constatação indica que, mais do que em qualquer outra atividade, o planejamento é imprescindível para o setor florestal.

Neste sentido, muitos aspectos têm sido observados no planejamento, como a utilização de modelos biométricos com enfoque à prognose do crescimento e da produção da floresta, os quais consistem em técnicas que utilizam uma série de funções matemáticas de forma conjunta e seqüencial com o objetivo de estimar os valores das variáveis envolvidas no crescimento e na produção dos povoamentos florestais. Também, são utilizados métodos de avaliação econômico-financeira e de planejamento e otimização, que consideram os critérios econômicos para a avaliação de projetos de

investimentos, envolvendo engenharia econômica e a programação matemática (OLIVEIRA, 1995).

No momento da colheita, muitas vezes, se faz necessária a consideração de condições que assegurem a dispersão da colheita, seja esta necessidade baseada em aspectos ambientais ou para atender a exigências legais que regulamentam a extensão da floresta que pode ser colhida no período.

Para assegurar estas condições precisa-se considerar no modelo de planejamento restrições de adjacência. Este tipo de restrição exige o uso de variáveis binárias para que se possa garantir a adjacência. Para resolver este problema pode-se optar entre os métodos exatos de programação linear inteira binária ou a utilização de técnicas heurísticas.

O Algoritmo Branch and Bound e o algoritmo de Balas são os métodos mais utilizados para a obtenção de uma solução exata para problemas de programação linear inteira.

Como o problema de adjacência vem sendo progressivamente reconhecido como fundamental dentro de um modelo de planejamento florestal, vários pesquisadores têm estudado técnicas heurísticas que se adaptem a este tipo de problema. Dentre as heurísticas que já foram aplicadas pode-se citar: Programação Inteira Monte Carlo (Monte Carlo Integer Programming), Simulated Annealing, Busca Tabu e Algoritmos Genéticos.

Claramente, é importante procurar o melhor caminho para resolver problemas de planejamento florestal com restrições de adjacência. Os softwares comerciais de otimização têm melhorado drasticamente.

Problemas que eram difíceis de ser resolvido com um algoritmo exato, hoje podem ser resolvidos de forma relativamente fácil.

Os softwares de otimização estão sendo melhorados continuamente e estes são muito flexíveis e podem manusear uma ampla variedade de formulações de problemas. Aspectos estes que mostram uma vantagem para sua utilização na resolução de problemas de planejamento florestal tendo em vista que este tipo de planejamento envolve um número muito grande de variáveis e restrições.

### **1.1 Objetivo do Trabalho**

O que se propõe neste trabalho é abordar o problema da consideração de restrições de adjacência em um modelo de planejamento florestal com restrições de demanda por determinados produtos e restrições de corte, buscando maximizar a receita da empresa florestal.

O objetivo para este trabalho é formular as restrições de adjacência aplicadas a talhões vizinhos em um mesmo período e em períodos consecutivos, com a intenção de evitar-se a erosão e também evitar que áreas de grandes extensões sejam devastadas em uma mesma região.

Também se pretende utilizar as restrições de adjacência para que as mesmas evitem que em um mesmo período talhões muito distantes sejam selecionados para colheita, considerando-se que a colheita exige manutenção de estradas e deslocamento de maquinários e funcionários até o local, não sendo assim interessante que em um mesmo período colha-se talhões distantes uns dos outros, pois, este

fato, acarretaria custos mais elevados. Ao nos referirmos a talhões distantes não se pretende considerar apenas a distância entre os talhões, mas sim, além da distância, as condições em que se encontram as estradas de acesso aos mesmos ou outro impedimento que possa ser considerado pela empresa no momento da colheita.

Também é apresentada neste trabalho uma revisão sobre os métodos para a solução de modelos de Programação Linear Inteira Binária e métodos específicos para a formulação e resolução do modelo florestal com restrições de adjacência. Para resolver o problema de Planejamento Florestal apresentado neste trabalho será utilizado o software LINGO (Language for Interactive General Optimizer).

## **1.2 importância do Trabalho**

O planejamento das colheitas futuras é a principal atividade do gerenciador florestal tendo em vista que seu objetivo fundamental é a maximização da utilidade da floresta, e por questões ambientais e/ou táticas a dispersão da colheita é um ponto de grande interesse durante a fase de planejamento e, este aspecto é assegurado utilizando-se restrições de adjacência.

A utilização das restrições de adjacência no modelo de planejamento florestal ajudarão a atender aspectos legais na hora da colheita, como, por exemplo, obedecer ao decreto ambiental que regulamenta a abertura máxima permitida para a colheita em 600 hectares para uma faixa contínua no período considerado.

Também se pode ressaltar a importância ambiental do uso de restrições de adjacência uma vez que evitaremos que grandes áreas

de terra sejam devastadas em uma região fazendo-se assim um controle da erosão e preservando o habitat natural da região.

### **1.3 Limitação do Trabalho**

A utilização de restrições de adjacência em um modelo florestal aumenta muito o número de restrições e exige o uso de variáveis binárias no modelo, o que pode incrementar exageradamente a exigência computacional, tornando-o, em ocasiões, impossível de ser resolvido. O aumento no número de restrições é devido a estas serem impostas a cada talhão considerado no modelo e este fato pode acarretar um tempo computacional elevado para sua resolução.

### **1.4 Estrutura do trabalho**

No capítulo II deste trabalho são apontadas as utilizações de Pesquisa Operacional no Planejamento Florestal, algumas técnicas de análise de investimentos e também os dois principais modelos de Programação Linear para Planejamento Florestal. Ainda no capítulo II temos a descrição dos métodos exatos para solução de problemas de Programação Linear Inteira Binária.

Alguns trabalhos publicados com uso de restrições de adjacência aplicados ao planejamento florestal são apresentadas no capítulo III.

A formulação matemática das restrições de adjacência no planejamento florestal e as demais restrições envolvidas no modelo, são apresentados no capítulo IV.

No capítulo V é realizada a análise dos resultados obtidos nos testes realizados para avaliar a formulação das restrições de adjacência proposta neste trabalho.

Finalmente, o capítulo VI contém as conclusões e sugestões para trabalhos futuros.

## CAPÍTULO II

### 2. MODELOS DA PESQUISA OPERACIONAL PARA O PLANEJAMENTO FLORESTAL

A Pesquisa Operacional surgiu durante a 2ª Guerra Mundial na Inglaterra com o objetivo de analisar e dar soluções a problemas que escapavam da rotina bélica, tanto no plano tático como estratégico. Os primeiros estudos feitos referiam-se ao emprego eficiente do radar, o uso de canhões antiaéreos, táticas de bombardeio a submarinos, escoltas navais, etc. O sucesso desses grupos de estudo é que deu incentivo para que outros também começassem seus trabalhos, disseminando assim o uso da Pesquisa Operacional. Porém o marco definitivo para a consolidação da Pesquisa Operacional veio com a publicação de Dantzig, em 1947, do método simplex para a programação linear, método este que permanece em utilização até os dias de hoje (PUCCINI, 1990).

Apesar das técnicas de otimização terem várias aplicações na área florestal sua utilização em situações práticas foi inicialmente limitada. Essa limitação é devida à falta de equipamentos computacionais que resolvessem automaticamente os cálculos envolvidos nos algoritmos proporcionados pelo grande número de variáveis envolvidas em problemas reais desta natureza. WAGNER (1985) aponta que a construção de modelos é a essência da abordagem de Pesquisa Operacional. Construir um modelo ajuda o planejador a colocar as complexidades e possíveis incertezas que acompanham um problema de tomada de decisão dentro de uma estrutura lógica passível de uma análise abrangente. DYKSTRA (1984) identifica os seguintes tipos de modelos:

- a) Modelos icônicos: são representações físicas de um sistema real, como, por exemplo, as esculturas, as miniaturas, os mapas e as fotografias.
- b) Modelos analógicos: são aqueles modelos que representam um sistema “de forma análoga”, mas que não tem qualquer semelhança física com o sistema sendo representado. SPIECKER (1975) enfatiza que um modelo analógico é um equivalente físico da realidade que ele representa. Como exemplos podem ser citados os gráficos e fluxogramas.
- c) Modelos simbólicos: estes modelos são inicialmente concebidos na forma de pensamento abstrato de um sistema, e, na seqüência, materializados através do uso de símbolos como em fórmulas químicas ou equações matemáticas. Na Engenharia Florestal, exemplos clássicos são as equações de volume, de mortalidade, os modelos (funções) de crescimento e de produção, e os modelos para planejamento estratégico da produção florestal.

CARNIERI (1989) ressaltava que o sucesso de um modelo em pesquisa operacional para o planejamento florestal depende de seu nível de adequação à situação real. Esta “imitação da realidade” deve ser aceitável pelo planejador florestal, pois, caso contrário, o modelo matemático construído não conseguirá captar todas as variáveis importantes do trabalho. As variáveis envolvidas em problemas de planejamento florestal têm natureza diferente, algumas são biológicas, algumas econômicas e sociais e outras são de origem ambiental, sendo que geralmente estas variáveis são relacionadas e qualquer mudança em uma delas terá influência sobre as outras.

Com o desenvolvimento da informática a Pesquisa Operacional também teve grandes avanços, pois os avanços tecnológicos possibilitaram o uso de computadores com maior capacidade e velocidade

e este fato facilitou a resolução de problemas com um número maior de restrições e variáveis, abrindo desta forma um leque maior de aplicações para a Pesquisa Operacional, tornando-a assim cada vez mais utilizada.

O processo de planejamento em uma empresa florestal é uma atividade de vital importância, pois um dos objetivos dessa empresa é a produção de madeira com fins econômicos, sociais ou ambientais envolvendo atividades que devem ser desenvolvidas em um certo período de tempo e as disponibilidades de recursos físicos e financeiros referentes a este período (BUONGIORNO, 1987). Segundo WARE e CLUTTER (1977) o objetivo do gerente florestal é maximizar a utilidade da floresta.

Para a atividade florestal, a base para o planejamento está na determinação dos regimes de manejo, principalmente no que se refere à idade para o corte final, ou para a rotação do povoamento, que possibilite a máxima rentabilidade.

Para efetuar a avaliação econômica no planejamento florestal, normalmente são utilizadas as técnicas do Valor Presente Líquido (VPL) e Valor Presente Líquido Anualizado (VPLa) ( OLIVEIRA, 1998).

Estas técnicas serão descritas a seguir:

## **2.1 Técnicas de análise de investimentos**

SOUZA e CLEMENTE (1999) enfatizam que um investimento, para a empresa, é um desembolso que é feito visando gerar um fluxo de benefícios futuros, usualmente superiores a um ano. Hoje, em função da própria dinâmica dos negócios, as técnicas de análise de investimentos estão sendo usadas tanto para investimentos de porte, associados a longos horizontes de planejamento, como é o caso do planejamento florestal, como também para operações de curto prazo, como por

exemplo, nas decisões rotineiras sobre compras a vista versus compras a prazo.

A decisão de fazer investimento de capital é a parte de um processo que envolve a geração e a avaliação das diversas alternativas que atendam às especificações técnicas dos investimentos. Depois de relacionadas às alternativas viáveis tecnicamente é que se analisará quais delas são atrativas financeiramente. É nessa última parte desse processo que são empregadas as técnicas de análise de investimentos. Os Projetos de Investimentos serão representados por um Fluxo de Caixa tal como apresentado a seguir:

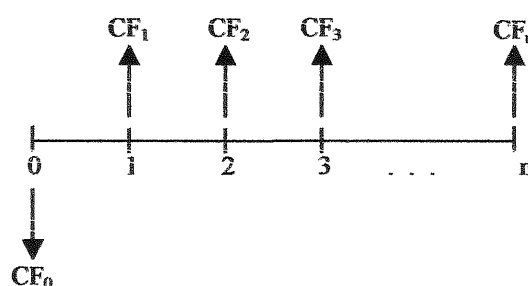


Figura 2.1.1: Representação esquemática de um fluxo de caixa.

As técnicas de análise de investimentos podem ser subdivididas em dois grandes grupos, quais sejam: técnicas que servem para selecionar projetos e técnicas que servem para gerar indicadores adicionais para projetos já selecionados. Na primeira categoria estão os chamados Métodos Robustos de Análise de Investimentos, que são: Método do Valor Presente Líquido (VPL) e Método do Valor Presente Líquido Anualizado (VPLa).

### 2.1.1 Método do Valor Presente Líquido

O método do Valor Presente Líquido (VPL), com, certeza, é a técnica robusta de análise de investimento mais conhecida e mais utilizada. O Valor Presente Líquido, como o próprio nome indica, nada

mais é do que a concentração de todos os valores esperados de um fluxo de caixa na data zero. Para tal, usa-se como taxa de desconto a Taxa Mínima de Atratividade da empresa (TMA).

Assim, pra o fluxo de caixa da figura 2.2.1, o Valor Presente Líquido seria calculado da seguinte forma:

$$VPL = \sum_{j=0}^n \frac{CF_j}{(1+i)^j}$$

Onde  $CF_j$  representa o valor do fluxo de caixa no tempo  $j$  e  $i$  é a TMA adotada pela empresa.

### 2.1.2 Método do VPL anualizado

O Método do Valor Presente Líquido Anualizado (VPLa) é mais uma variação do Método do Valor Presente Líquido. Enquanto no Método do VPL todos os valores do fluxo de caixa são concentrados na data zero, no Método VPLa o fluxo de caixa representativo do projeto de investimento é transformado em uma série uniforme. O procedimento para esta transformação é bastante simples, porém um pouco trabalhoso.

$$VPLa = VPL \cdot \frac{i(1+i)^n}{(1+i)^n - 1}$$

Vários autores utilizaram o VPL para fazer a análise financeira no planejamento florestal, entre eles pode-se citar: OLIVEIRA et al (1998) em seu artigo "Sistema para simulação do crescimento e da produção de *Pinus taeda* L.E. e a avaliação econômica de regimes de manejo" e SCOLFARO (1990) em sua tese de doutorado "Sistema Integrado para predição e análise presente e futura do crescimento e produção, com

otimização de remuneração de capitais, para *Pinus caribaea* var. *hondurensis*.

BRAZ (2001) utilizou em sua dissertação de mestrado "Um modelo de programação linear para garantia do rendimento sustentado em pequena propriedade na floresta tropical" um modelo matemático que auxilia no planejamento e distribuição dos talhões em uma pequena floresta tropical, para que se pudesse garantir um fluxo constante de madeira com potencial comercial para a viabilidade da pequena propriedade.

HOSOKAWA (1982) destaca que planejar a produção de uma floresta significa definir os objetivos pretendidos, ou seja, os retornos de investimentos e prever, em termos de longo prazo, a produção total, incluindo todos os cortes a serem realizados no povoamento, quais sejam: desbastes, cortes seletivos, cortes para regeneração, corte final, etc. Trata-se do ordenamento de uma floresta tanto em termos de otimização da produção de madeira, da produção financeira, quanto da produção de benefícios indiretos (sociais e ecológicos) a um custo mínimo, sejam eles custos de produção, custos sociais, ou custos ecológicos.

Com o uso da Programação Linear para a resolução de Problemas de Planejamento Florestal, vários modelos de planejamento florestal já foram desenvolvidos e aplicados. A seguir serão apresentados os dois modelos básicos de Planejamento Florestal desenvolvidos por JOHNSON e SCHEURMAN (1997).

## **2.2 Modelos de Programação Linear para Planejamento Florestal**

Existem dois modelos básicos de Programação Linear para otimizar a produção, a colheita de madeira e o investimento, ao longo de um período de planejamento, os quais foram publicados no periódico *Forest Science*, em uma monografia de JOHNSON e SCHEURMAN (1977). Os dois

modelos são definidos como modelo I e modelo II e a diferença básica entre estes dois modelos está na forma de se definir as atividades.

No modelo I uma atividade é representada por ações tomadas sobre as unidades de manejo durante o horizonte de planejamento. No modelo II, uma atividade é representada por ações que atuam sobre talhões de uma classe de idade existente durante o horizonte de planejamento. A forma matemática destes dois modelos é apresentada sob a forma de uma função objetivo e restrições de área disponíveis em um determinado período.

No modelo I são definidas as unidades e regimes de manejo e cada classe de idade no 1º período do horizonte de planejamento forma uma unidade de manejo (ou talhão) que será conservada por todo horizonte de planejamento. O modelo I, mais simples, pode ser representado matematicamente por:

$$\text{Max } \sum_{i=1}^{NE} \sum_{j=1}^{NR_i} c_{ij} x_{ij}$$

$$\text{s.a. } \sum_{j=1}^{NR_i} x_{ij} = \text{ÁREA}_i \quad i = 1, 2, \dots, NE$$

$x_{ij} \geq 0$  onde:

NE = número de unidades de manejo

NR<sub>i</sub> = número de regimes de manejo associados a unidade i

ÁREA i = área da unidade i

c<sub>ij</sub> = valor líquido presente da receita por hectare, da unidade i que segue o regime de manejo j.

x<sub>ij</sub> = número de hectares da unidade i manejada pelo regime j.

No modelo II uma atividade refere-se a um conjunto de ações que podem ocorrer em uma determinada área desde o seu plantio até seu corte raso ou inventário final. Matematicamente o modelo II é representado por:

$$\text{Max} \sum_{j=1}^H \sum_{i=-M}^{j-Z} c_{ij} x_{ij} + \sum_{i=-M}^H d_{iH} w_{iH}$$

$$\text{s.a.} \sum_{j=1}^H x_{ij} + w_{iH} = \text{ÁREA}_i \quad i = -M, \dots, 0$$

$$\sum_{k=j+Z}^H x_{jk} + w_{jH} = \sum_{i=-M}^{j-Z} x_{ij}$$

$$x_{ij} \geq 0 \quad w_{iH} \geq 0 \quad \text{onde:}$$

$x_{ij}$  ( $x_{jk}$ ) = número de hectares plantados no período  $i$  ( $j$ ) e cortados no período  $j$  ( $k$ ).

$w_{iH}$  ( $w_{jH}$ ) = número de hectares plantados no período  $i$  ( $j$ ) e deixados para inventário final no período  $H$ .

$M$  = número de períodos anteriores ao período zero de planejamento, representando a classe de idade mais velha presente no início do horizonte de planejamento.

$H$  = período final de planejamento.

$Z$  = número mínimo de períodos entre colheitas.

$c_{ij}$  = valor presente líquido por hectare das unidades plantadas no período  $i$  e colhidas no período  $j$ .

$d_{iH}$  = valor presente líquido por hectare das unidades plantadas no período  $i$  e deixadas para inventário final no período  $H$ .

$\text{ÁREA}_i$  = número de hectares no primeiro período do horizonte de planejamento plantados no período  $i$ ,  $i = -M, \dots, 0$ .

O primeiro conjunto de restrições no modelo II afirma que os hectares cortados em cada classe de idade do período 1 mais os hectares deixados como inventário final para aquela classe de idade devem ser igual ao total de hectares nesta classe de idade. E o segundo conjunto de restrições afirma que os hectares cortados e plantados no período  $j$  e colhidos no período  $H$  somados aos hectares plantados no período  $j$  e deixados para inventário final no período  $H$  é igual ao total de hectares cortados e imediatamente plantados no período  $j$ .

O modelo I preserva intactas as áreas do início do planejamento e isto já não ocorre no modelo II. Desta forma o modelo I permite identificar com mais clareza a área manejada na floresta, o que justifica sua maior aplicabilidade.

Entre os trabalhos que utilizam o modelo I podem-se citar BARROS e WEINTRAUB (1982) que utilizam este modelo para desenvolver um sistema de planejamento para indústria florestal integrada verticalmente, CARNIERI (1989) em sua tese de doutorado e CARNIERI et al (1991) utilizou o modelo I para desenvolver um sistema de planejamento florestal.

Também SCOLFORO (1990) em sua tese de doutorado, onde desenvolveu um sistema integrado para análise de crescimento e produção para Pinus com otimização de remuneração de capitais. E, mais recentemente, VOLPI (1997), em sua tese de doutorado, analisou o impacto de perturbações estocásticas em um modelo de planejamento florestal.

Muitos problemas podem ser expressos como problemas de otimização com restrições, sendo a programação linear uma das mais importantes e mais usadas técnicas para problemas deste tipo. (ZIONTS - 1974).

A programação inteira é a denominação dada a problemas de programação linear em que as variáveis só podem assumir valores inteiros e um caso particular de programação inteira é aquele em as variáveis só assumem valores 0 ou 1. Neste caso, o modelo é denominado programação zero-um ou binária.

A programação inteira desenvolveu-se em contextos de programação linear. Pode-se pensar em resolver um problema de programação inteira como um problema ordinário de programação linear e arredondar o valor da solução ótima para o inteiro mais próximo. Porém este procedimento é inviável quando os números envolvidos no problema são baixos, gerando assim uma incerteza no arredondamento, por exemplo, se uma variável tem valor 0,65 ela deve ser arredondada para zero ou um?

Intuitivamente pode-se pensar que um problema de programação inteira é muito mais simples de se resolver que um problema de programação linear com variáveis contínuas, mas a realidade é outra. Um bom computador resolve em poucos minutos um problema de programação linear com muitas variáveis e restrições, enquanto um problema de programação inteira poderá não ser resolvido em um tempo aceitável. O tempo considerado aceitável varia de problema para problema, por exemplo, um problema que deva ser resolvido todos os dias para que sua solução seja utilizada, não pode levar mais que alguns minutos para ser resolvido. Enquanto, para um problema de planejamento florestal, que é um planejamento em longo prazo, um tempo de 4 horas para sua resolução, por exemplo, é considerado aceitável uma vez que o problema é resolvido uma vez e a utilização de sua resposta ocorrerá no decorrer do ano. Desta forma, pode se observar que o tempo aceitável para a resolução de um problema dependerá de sua natureza.

A programação inteira ampliou enormemente a variedade de problemas onde a programação linear pode ser aplicada, e isto explica o desenvolvimento de técnicas específicas para sua resolução.

### **2.3 Técnicas para resolução de PPLI**

Entre as técnicas de resolução de um problema de programação inteira podemos citar os métodos exatos e os métodos heurísticos. Quando precisamos optar entre o uso de métodos exatos ou heurísticos temos um dilema entre eficiência e eficácia. A eficiência significa resolver um problema, encontrando uma solução ótima sem se preocupar com o tempo que isso leva. Já a eficácia significa resolver um problema, isto é, encontrar uma solução, não necessariamente a melhor, dentro de um tempo razoável. O tempo pode ser uma restrição na busca de uma solução. (ANDRADE, 2002)

A heurística é uma técnica que melhora a eficácia de um processo de busca, possivelmente, sacrificando a obtenção da solução ótima. Ela é como um guia de turismo. É válida no sentido que aponta para direções geralmente interessantes; e é imprópria no sentido de que deixa de fora pontos de interesse para determinados indivíduos. Usando-se uma boa heurística, pode-se esperar obter boas soluções para problemas difíceis, como o do caixeiro viajante. (LOESCH e HEIN, 1989).

Entre os métodos exatos para resolver um problema de programação inteira podemos citar o Algoritmo Branch and Bound e o algoritmo de Balas, descritos a seguir.

#### **2.3.1 O Algoritmo “Branch and Bound”**

Branch and Bound é uma técnica bastante utilizada para a resolução exata de problemas combinatórios NP-hard. É um método simples que foi

publicado em um artigo de LAND e DOIN (1960). Este requer o uso do método simplex e pode resolver problemas de programação inteira e mista. Neste método primeiro é alcançada uma solução ótima não inteira e em seguida efetuam-se partições no espaço das soluções e conclui-se a otimalidade da solução usando limites calculados ao longo da enumeração das soluções (ZIONTS, 1974).

O princípio do Branch and Bound é a enumeração de todas as soluções viáveis de um problema de otimização. Um ramo da árvore de enumeração define um subconjunto do conjunto de todas as soluções viáveis do problema original onde cada elemento do subconjunto satisfaz este ramo.

RENDER e STAIR (1997) dizem que o Branch and Bound é um algoritmo que busca por uma solução ótima examinando somente uma parte do número total de possíveis soluções. Ele trabalha quebrando o espaço de soluções viáveis em subproblemas menores até que uma solução ótima seja alcançada. Para cada subproblema gerado o custo total ou lucro total é calculado. Subproblemas com pior custo ou lucro são descartados até que não se possam criar mais subproblemas.

BLAZEWICZ et al. (1996) resumiram o significado do Branch and Bound: Branching (ramificação) é o procedimento de particionar um problema grande em dois ou mais subproblemas, geralmente mutuamente excludentes. Bounding (poda) calcula um limite inferior para o valor da solução ótima para cada subproblema gerado no processo de branching. Um nó pode ser eliminado não somente baseado no limite inferior, mas também por meio do critério de eliminação proporcionado por propriedades de dominância ou condições de viabilidade desenvolvidas para um dado problema.

A escolha de um nó a partir do conjunto de nós gerados é resultado da estratégia de busca adotada. A técnica Branch and Bound trabalha basicamente usando uma das duas estratégias, busca em largura ou busca em profundidade. Na busca em largura o processo de ramificação saí de um ramo para outro na árvore de busca, enquanto, na busca em profundidade, primeiro prossegue-se até o nível mais baixo por algum caminho para encontrar uma solução e então refaz-se aquele caminho para cima até o primeiro nível com nós ativos e assim por diante. Na busca em largura tende-se a construir uma grande lista de nós ativos e, na busca em profundidade mantém-se relativamente poucos nós na lista a qualquer momento.

Para construir um algoritmo de Branch and Bound para um dado problema, deve-se decidir sobre:

- i. O procedimento de ramificação e a estratégia de busca.
- ii. O procedimento de limitação ou critério de eliminação.

### 2.3.1.1 Descrição do algoritmo Branch and Bound

Seja o problema de programação linear inteira mista (PPLIM) apresentado na forma abaixo (MAYERLE, 2001):

$$\begin{aligned}
 &\mathbf{Max} \quad \mathbf{c}^T \mathbf{x} \\
 &\mathbf{s.a.} \quad \mathbf{Ax} = \mathbf{b} \\
 &\quad \mathbf{x} \geq \mathbf{0} \\
 &\quad \mathbf{x} = (\mathbf{x}_c, \mathbf{x}_i)^t \\
 &\quad \mathbf{x}_c = \text{variáveis contínuas} \\
 &\quad \mathbf{x}_i = \text{variáveis inteiras}
 \end{aligned}$$

**Passo 1:** Gere uma lista de PPL's inicialmente vazia. Resolva o PPLIM desconsiderando as restrições de integralidade. Se este problema não tem solução viável, então **PARE**. O PPLIM também não terá solução viável. Em caso contrário, inclua este PPL na lista, estes valores proporcionam um limite superior inicial.

**Passo 2:** Encontrar qualquer solução viável que preencha as condições de integralidade e usar estes valores como um limite inferior. Normalmente pode-se encontrar estes valores pelo arredondamento da solução encontrada no passo 1.

**Passo 3:** Escolha, entre os PPL's da lista, aquele cujo valor da solução ótima seja o máximo. Se mais de um problema atender esta condição, desempate de qualquer forma, mas sempre a favor de soluções que satisfaçam as condições de integralidade do PPLIM.

**Passo 4:** Retire o problema escolhido da lista e, se a solução satisfizer as condições de integralidade, **PARE**. A solução ótima do PPLIM foi encontrada. Em caso contrário, tome uma variável que deva ser inteira e que não satisfaz a condição de integralidade e gere dois novos PPL's, pela agregação das seguintes restrições adicionais:

$$\text{Problema 1} \quad x_j \leq \lfloor x_j^* \rfloor$$

$$\text{Problema 2} \quad x_j \geq \lceil x_j^* \rceil$$

Resolva cada PPL gerado.

A - se o problema tem solução inviável descarte-o.

B - se o problema tem solução viável mas não inteira, inclua-o na lista.

C - se o problema tem solução inteira viável, examine o valor da função objetivo. Se este valor é igual ao limite superior, uma solução ótima foi alcançada. Se ele não é igual ao limite superior, mas é maior que o limite inferior, adote-o como novo limite inferior e vá ao passo 5. Finalmente se ele é menor que o limite inferior descarte-o.

**Passo 5:** Se a lista estiver vazia, então **PARE**, pois o PPLIM não tem solução viável. Em caso contrário, volte ao passo 3.

MCDILL e BRAZE (2001) consideram o algoritmo Branch and Bound um método geral para obter uma solução exata para problemas de Programação Linear Inteira Mista, e afirmam que até recentemente somente problemas pequenos podiam ser resolvidos com este algoritmo e que sua capacidade para resolver problemas nunca foi minuciosamente avaliado. Por isso, em seu artigo, procuram avaliar a capacidade de um software implementado com Branch and Bound em termos de tamanhos e tipos de problemas de planejamento florestal com restrições de adjacência que podem ser resolvidos dentro de um tempo razoável, fixado em mais ou menos quatro horas.

O software avaliado por eles é o CPLEX 6.5 que é uma divisão da [ILOG](#) responsável pelo desenvolvimento de software para programação matemática e otimização de recursos de larga escala. Suas bibliotecas (*solvers*) para resolução de problemas de programação linear, inteira mista e quadrática, são reconhecidas por sua alta performance e confiabilidade. Ele contém *solvers* para problemas de otimização segundo os algoritmos SIMPLEX (*primal simplex*, *dual simplex* e *network simplex*), problemas de variáveis inteiras mistas geral ou binária. Esta variedade de opções proporciona ao usuário grande

flexibilidade para resolução de problemas. O CPLEX pode ser utilizado de forma interativa ou programada. A forma interativa (linha de comando) oferece facilidades tais como geração de relatórios, utilitários para leitura e escrita de arquivos, controle de mensagens, capacidade de revisão interativa, estrutura de comandos simples com *help on-line* e outras. A forma programada permite escrever programas (por exemplo, em C/C++) que constroem um problema e fazem chamadas às bibliotecas CPLEX para resolvê-lo.

A formulação de problemas com variáveis de decisão binárias permite incorporar decisões do tipo SIM / NÃO ao modelo. Essas variáveis são restritas a dois valores  $x_j = 1$  se a decisão for SIM e  $x_j = 0$  caso contrário. Para resolução deste tipo de problema pode-se utilizar o Algoritmo de Balas descrito a seguir.

### 2.3.2 O Algoritmo de Balas

O algoritmo de Balas (E. BALAS, 1965) é um algoritmo típico de busca em árvore binária, que consiste em igualar a zero tantas variáveis quanto permitam as restrições para dar preferência ao uso de variáveis que tenham subíndices menores quando necessário igualar a um.

#### 2.3.2.1 Descrição do Algoritmo de Balas

Seja o Problema de Programação Linear Binária (PPLB) na forma padrão, segundo MAYERLE (2001):

$$\text{Max} \sum_{j=1}^n c_j x_j$$

$$\text{s.a. } \sum_{j=1}^n a_{ij}x_j \leq b_i \quad \forall i = 1, \dots, m$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n$$

no qual  $c_1 \geq c_2 \geq c_3 \geq \dots \geq c_n \geq 0$ .

No caso do problema não se adaptar a essa forma padrão, o mesmo deverá sofrer as seguintes transformações:

a) efetue a substituição  $x_j = 1 - x_j'$   $\forall c_j < 0$ .

b) troque cada uma das restrições do tipo  $\sum a_{ij}x_j = b_i$  por uma restrição do tipo  $\sum a_{ij}x_j \leq b_i$  e outra do tipo  $\sum a_{ij}x_j \geq b_i$ .

c) troque cada uma das restrições do tipo  $\sum a_{ij}x_j \geq b_i$  por uma restrição  $-\sum a_{ij}x_j \leq -b_i$ .

d) reordene as variáveis do PPLB de modo que  $c_1 \geq c_2 \geq c_3 \geq \dots \geq c_n$ .

O algoritmo consiste em um processo de busca em árvore binária, na qual a cada nó da árvore corresponde uma solução parcial do PPLB, cujos valores das  $k$  primeiras variáveis são conhecidas. A fim de se verificar a possibilidade de existência de uma solução viável e ótima dois testes deverão ser realizados. Estes testes são expostos a seguir:

#### i. Teste de Otimalidade

O teste de otimalidade considera o conhecimento prévio de uma solução ótima temporária, cujo valor da função objetivo é  $Z_{\text{ótimo}}$ . Considerando que na solução parcial as  $k$  primeiras variáveis são

conhecidas, uma estimativa do valor da função objetivo para uma solução completa deriva desta solução parcial e poderá ser feita através de:

$$Z_{\text{estimado}} = \sum_{j=1}^k c_j x_j + \sum_{j=k+1}^n c_j$$

Se  $Z_{\text{estimado}}$  for superior ao valor de  $Z_{\text{ótimo}}$ , então é possível que se encontre uma solução ótima a partir dessa solução parcial. Em caso contrário, a melhor solução possível de ser encontrada não será melhor que a solução ótima temporária disponível.

### ii. Teste de Viabilidade

O teste de viabilidade, por sua vez, considera que para cada restrição deve ser satisfeita a seguinte condição:

$$\sum_{j=1}^k a_{ij} x_j + \sum_{j=k+1}^n a_{ij} x_j \leq b_i$$

ou

$$\sum_{j=1}^k a_{ij} x_j \leq b_i - \sum_{j=k+1}^n \min\{0; a_{ij}\}$$

Então, se para alguma restrição:

$$\sum_{j=1}^k a_{ij} x_j > b_i - \sum_{j=k+1}^n \min\{0; a_{ij}\}$$

Conclui-se que não existirão valores para as variáveis ainda não conhecidas que permitam a obtenção de uma solução que satisfaça a restrição.

No caso de falha de um dos dois testes, é necessário rever os valores atribuídos as  $k$  primeiras variáveis.

O algoritmo propriamente dito é descrito a seguir:

**Passo 0.** Escreva o PPLB em sua forma padrão. Faça  $Z_{\text{ótimo}} = -\infty$  e  $k = 1$ ;

**Passo 1.** Faça  $x_k = 1$ . verifique se:

$$Z_{\text{estimado}} = \sum_{j=1}^k c_j x_j + \sum_{j=k+1}^n c_j > Z_{\text{ótimo}} \quad e$$

$$\sum_{j=1}^k a_{ij} x_j > b_i - \sum_{j=k+1}^n \min\{0, a_{ij}\} \quad i = 1, 2, \dots, m$$

Em caso de sucesso ir ao passo 3.

**Passo 2.** Faça  $x_k = 0$ . Verifique se:

$$Z_{\text{estimado}} = \sum_{j=1}^k c_j x_j + \sum_{j=k+1}^n c_j > Z_{\text{ótimo}} \quad e$$

$$\sum_{j=1}^k a_{ij}x_j > b_i - \sum_{j=k+1}^n \min\{b_i; a_{ij}\} \quad i = 1, 2, \dots, m$$

em caso de fracasso vá ao passo 4.

**Passo 3.** Faça  $k = k+1 \leq n$ , volte ao passo 1. Senão, uma solução completa, melhor que a solução ótima temporária, foi encontrada. Guarde esta solução como sendo a nova solução ótima temporária. Atualiza o valor do  $Z_{\text{ótimo}}$ .

**Passo 4.** Backtrack. Encontre  $K = \{j \mid x_j = 1 \text{ e } 0 < j < k\}$ . Se  $k = \emptyset$ , então PARE; a solução ótima temporária é a solução ótima do problema. Em caso contrário, determine  $k = \{j \mid j \in K\}$  e retorne ao passo 2.

No próximo capítulo apresentaremos trabalhos publicados com a utilização de restrições de adjacência no planejamento florestal.

### **CAPÍTULO III**

#### **3. APLICAÇÕES DE TÉCNICAS DESENVOLVIDAS PARA RESOLUÇÃO DE PROBLEMAS COM RESTRIÇÕES DE ADJACÊNCIA**

Quando se utiliza programação linear no planejamento florestal de longo e médio prazo, são necessárias algumas condições para assegurar a dispersão da colheita. O controle da dispersão da colheita deve ser feito para amenizar os efeitos da erosão, evitar grandes devastações e atender a aspectos legais no que diz respeito ao tamanho da colheita permitida. Estas condições são satisfeitas inserindo-se restrições de adjacência no modelo considerado, que irão excluir combinações de atividades indesejáveis em talhões adjacentes. Por exemplo, se considerarmos as unidades de trabalho delimitadas por polígonos, pode-se desejar que polígonos com lados comuns não sejam colhidos em um mesmo período, ou em um intervalo específico de tempo.

O problema da consideração de restrições de adjacência no gerenciamento florestal tem sido progressivamente reconhecido como fundamental para ações locais específicas. O problema de adjacência é relevante por diversas razões, algumas delas ambientais. Com a consideração de restrições de adjacência pode-se evitar que grandes áreas sejam devastadas mantendo assim um certo equilíbrio ambiental observando nesta região uma paisagem em mosaico.

Apresentaremos a seguir trabalhos publicados nos quais foram utilizadas restrições de adjacência.

### 3.1 Formulação de restrições de adjacência em modelos de otimização linear para planejamento tático

Em JONES, et al. 1991 apresenta-se uma formulação para restrições de adjacência com um número menor de restrições, comparando-se com a formulação explícita da mesma. Considera-se o número de restrições um dos problemas que limitam o uso de programação linear em planejamento florestal devido a necessidade de um número muito grande de restrições mesmo para um modelo de tamanho modesto.

Nesta abordagem foi utilizado um delineamento da área da floresta em forma de polígonos que representam as unidades de corte, ou talhões, sendo que o tamanho dos polígonos não excedem o tamanho de colheita permitida. O modelo de programação linear é formulado de modo que exclua a colheita de unidades vizinhas em um determinado intervalo de tempo.

A figura 3.1.1, abaixo, mostra um exemplo do delineamento da área da floresta:

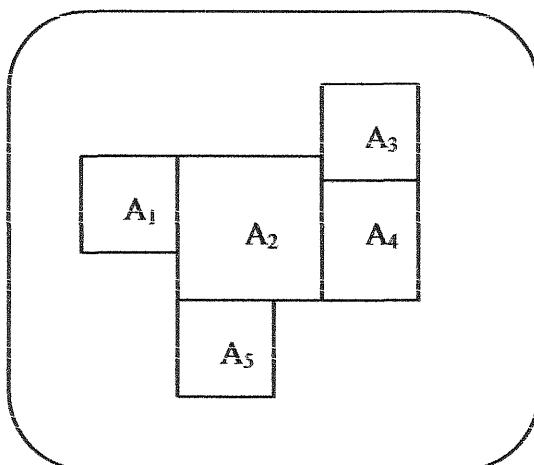


Figura 3.1.1: Delineamento da área da floresta

No exemplo, observamos uma floresta formada por cinco unidades de corte.

Nesta pesquisa foram consideradas restrições de adjacência tipo 1 e tipo 2 (restrições tipo-1, restrições tipo-2). As restrições tipo -1 são definidas em polígonos mutuamente adjacentes, estas equivalem a formulação explícita das restrições, uma vez que para cada unidade será exigida pelo menos uma restrição. As restrições de adjacência tipo-2, são a junção de duas ou mais restrições tipo-1 em uma única restrição.

Para as restrições tipo-1 devemos ter a soma das variáveis de decisão menor ou igual a um, da seguinte forma:

Restrição tipo-1:

$$(3.1.1) \quad \sum_{i \in A} Y_i \leq 1, \text{ onde } A \text{ é o conjunto de polígonos mutuamente}$$

adjacentes e  $Y_i$  são as variáveis binárias tal que:

$$Y_i = \begin{cases} 1 & \text{se a unidade de corte } i \text{ será selecionada} \\ 0 & \text{caso contrário} \end{cases}$$

Para definir o conjunto A, procurou-se identificar os polígonos com maior número de fronteiras comuns primeiro, para depois os polígonos com menos fronteiras comuns buscando desta forma preservar a adjacência com o menor número possível de restrições tipo 1.

No exemplo mostrado na figura 3.1.1 temos 3 conjuntos de unidades mutuamente adjacentes que são  $\{A_1, A_2\}$ ,  $\{A_2, A_5\}$  e  $\{A_2, A_3, A_4\}$ . Para este caso observamos que  $Y_3$  e  $Y_4$  já estão incluídos no conjunto  $\{Y_2, Y_3, Y_4\}$  não precisando inserir mais este conjunto.

Para cada um destes conjuntos teremos uma restrição tipo -1, da seguinte forma:

$$\begin{array}{rcl} Y_1 + Y_2 & & \leq 1 \\ Y_2 & + Y_5 & \leq 1 \\ Y_2 + Y_3 + Y_4 & & \leq 1 \end{array}$$

Para usar as restrições de adjacência tipo-1 impedindo a colheita de unidades vizinhas em um mesmo período, se faz necessário que cada unidade com fronteiras comuns apareçam em pelo menos uma restrição tipo-1.

Utilizam-se as restrições tipo-2 para reduzir o número de restrições de adjacência do modelo.

As restrições tipo-2 são definidas sob determinadas condições que devem ser satisfeitas pelos conjuntos de inequações tipo 1 que serão condensadas em uma inequação tipo 2. Este conjunto de restrições tipo 1 será designado por R. As unidades de corte pertencentes ao conjunto R serão divididas em três subconjuntos denominados E, P e G.

Para selecionarmos os conjuntos de polígonos mutuamente adjacentes que farão parte do conjunto R as seguintes condições devem ser satisfeitas:

$$(3.1.2) \quad \sum_{i \in E} Y_i + \sum_{i \in P} Y_i \leq 1$$

$$(3.1.3) \quad \sum_{i \in E} Y_i + \sum_{i \in G} Y_i \leq 1, \text{ sendo que a condição 3.1.3 é opcional, só}$$

deverá ser considerada caso o conjunto G, definido a seguir, seja não vazio.

O subconjunto E é o conjunto de unidades presentes em todas inequações de R, o subconjunto P é o conjunto de (r-1) unidades (se G é não vazio), onde r é o número de inequações pertencentes a R, sendo que cada membro de P está presente em somente uma inequação de R e G é o conjunto de uma ou mais unidades presentes em R, sem estar presente em E ou P.

Todos as unidades são incluídas nos conjuntos E, P ou G de modo que:  $E \cap P \cap G = \emptyset$  e E e P sejam não vazios.

No exemplo citado consideremos, inicialmente, o conjunto R como sendo as três restrições tipo -1 formuladas:

$$\begin{array}{rcl} Y_1 + Y_2 & & \leq 1 \\ Y_2 & + Y_5 & \leq 1 \\ Y_2 + Y_3 + Y_4 & & \leq 1 \end{array}$$

Sendo E o conjunto das unidades presentes em todas as restrições tipo -1 de R, observamos que somente a unidade  $A_2$  satisfaz esta condição, desta forma temos:  $E = \{A_2\}$ .

O subconjunto P deve conter (r -1) ou r unidades que devem estar presentes em somente uma restrição tipo -1 do conjunto R, as unidades de P não podem pertencer a uma mesma restrição de R, desta forma:

$P = \{A_1, A_5\}$ ,  $A_1$  está presente somente na primeira restrição e  $A_5$  somente na última restrição.

O conjunto P foi composto por duas unidades o que corresponde a (r-1) elementos, portanto o subconjunto G será não vazio.

O subconjunto G deve conter unidades que não estão presentes em E ou P, assim G será composto por  $A_3$  e  $A_4$  presentes na primeira restrição de R.

Definidos os conjuntos  $E = \{A_2\}$ ,  $P = \{A_1, A_5\}$  e  $G = \{A_3, A_4\}$ , verificamos se eles satisfazem as condições (4.2) e (4.3) apresentadas anteriormente.

$$(3.1.2) \quad \sum_{i \in E} Y_i + \sum_{i \in P} Y_i \leq 1$$

$$\rightarrow Y_2 + Y_1 \leq 1$$

$$\rightarrow Y_2 + Y_5 \leq 1$$

Nesta condição cada elemento de P é considerado separadamente, pois os elementos de P não pertencem a uma mesma inequação, assim podemos selecionar qualquer combinação entre eles, porque os mesmos não são vizinhos.

A condição (3.1.2) é satisfeita uma vez que não se deseja selecionar as unidades  $A_1$  e  $A_2$  ou  $A_5$  e  $A_2$  simultaneamente.

Como o conjunto G é não vazio deve-se verificar a condição (3.1.3) como segue:

$$(3.1.3) \quad \sum_{i \in E} Y_i + \sum_{i \in G} Y_i \leq 1$$

$$\rightarrow Y_2 + Y_3 + Y_4 \leq 1$$

Neste caso considera-se o somatório dos elementos de G porque eles pertencem a uma mesma inequação, o que significa que eles contêm

fronteiras comuns, desta forma não devem ser selecionados simultaneamente.

A condição (3.1.3) também foi satisfeita, pois caso a unidade  $A_3$  seja selecionada, não devemos selecionar as unidades  $A_4$  e  $A_2$ , conforme desejado.

Sendo as duas condições satisfeitas as restrições tipo -1 contidas no conjunto R escolhido serão substituídas por uma restrição tipo-2 na forma:

Restrição tipo-2:

$$(3.1.4) \quad (2r - 1) \sum_{i \in E} Y_i + \sum_{i \in P} Y_i + r \sum_{i \in G} Y_i \leq (2r - 1), \text{ onde } r \text{ é o número}$$

de inequações do conjunto R.

Desta forma as restrições tipo-1 de R serão substituídas pela seguinte restrição tipo-2:

$$5Y_2 + Y_1 + Y_5 + 3(Y_3 + Y_4) \leq 5$$

Formular as restrições de adjacência tipo-2 para uma área requer o desenvolvimento das restrições tipo-1 para esta área. Para obter um melhor resultado é necessário que as fronteiras comuns sejam incluídas no menor número possível de restrições tipo-1, então, devemos combinar as restrições tipo-1 dentro do conjunto R de modo que satisfaçam as condições (3.1.2) e (3.1.3) para se formular as restrições tipo-2. As restrições tipo-1 que não forem incluídas em nenhum conjunto R devem ser relacionadas ao longo das restrições tipo-2 para que todas as condições de adjacência sejam satisfeitas para essa área.

Analisando a inequação (3.1.4) verificamos que o conjunto E possui unidades presentes em todas as restrições tipo-1 do conjunto R, sendo

assim se uma unidade do conjunto E é selecionada nenhuma outra poderá ser. Pode-se observar nesta inequação que o coeficiente das inequações pertencentes ao conjunto E é  $(2r - 1)$  que é o mesmo valor do lado direito da desigualdade, então se uma unidade de E é selecionada sua contribuição será de  $(2r - 1)$  e nenhuma outra unidade presente na mesma inequação poderá ser selecionada sem ultrapassar o limite imposto pelo lado direito da desigualdade.

Todas as unidades do conjunto G estão presentes em uma mesma equação, então se pode selecionar apenas um desses projetos, o que é assegurado pela equação (3.1.4) no coeficiente de G que é r. Se, por exemplo, selecionarmos duas unidades de G sua contribuição será de  $2r$  o que ultrapassará o valor de  $(2r - 1)$  do lado direito da desigualdade.

As restrições tipo-1 não possuem mais de uma unidade do conjunto P, então é possível fazer combinações com as unidades de P, uma vez que seu coeficiente é 1. Assim, se forem selecionados todos as unidades de P teremos uma soma de r (no caso de G vazio) ou  $(r - 1)$ , sem ultrapassar o valor do lado direito da desigualdade. As inequações tipo-1 também não possuem unidades que pertençam a P e G simultaneamente, assim as restrições tipo-2 permitem a combinação entre as unidades do conjunto P e G. Caso sejam selecionadas todas as unidades de P teremos uma contribuição de  $(r - 1)$  na inequação o que permitirá a seleção de uma unidade do conjunto G com contribuição de r, sem ultrapassar o limite de  $(2r - 1)$ .

Segundo Jones et al, esta formulação pode reduzir substancialmente o número de restrições de adjacência requisitadas para um conjunto de polígonos, mas não indica o quanto se pode esperar de redução em uma área de planejamento completa. Foram realizados testes com 3 áreas, onde a área 1 tinha 27 polígonos (ou unidades), a área 2 com 112 polígonos e a área 3 com 279 polígonos.

Para a realização dos testes foram formuladas as restrições de adjacência tipo-1 para as fronteiras comuns utilizando a estratégia de identificar todas as combinações de polígonos com 4 fronteiras mutuamente adjacentes, em seguida três, duas e uma, esta estratégia auxilia a não esquecer nenhuma fronteira e também a não relacioná-la mais de uma vez sem que haja necessidade. Em seguida, foram desenvolvidas as restrições tipo-2 que satisfizessem as restrições de adjacência impostas para as áreas consideradas.

Os resultados dos testes realizados por JONES, foram apresentados como porcentagem do número de unidades e estão resumidos no quadro abaixo:

#### RESULTADOS DOS TESTES

Tipo da restrição	Área 1 (%)	Área 2 (%)	Área 3 (%)	Média (%)
Fronteiras	167	180	209	185
Tipo 1	100	109	118	109
Tipo 2	44	50	55	50

QUADRO 3.1.1: Resultado dos testes realizados por Jones et al.

A primeira linha do quadro mostra a porcentagem de fronteiras comuns entre polígonos adjacentes em relação ao número de polígonos. Na média o número de fronteiras comuns era 185% do número de polígonos, utilizando a formulação das restrições tipo-1 baixou para 109% e com as restrições tipo-2 passou para 50% do número de polígonos. Os resultados dos testes, realizados com três áreas, para esta formulação de restrições de adjacência mostraram que a redução pode ser significativa. Apesar desta redução no número de restrições, o que dificulta a utilização desta formulação abordada por JONES et al. (1990) é o trabalho despendido para identificar o conjunto R de restrições que satisfaça as

condições (3.1.2) e (3.1.3) impostas para assegurar a adjacência e também a identificação dos conjuntos E, P e G dentro do conjunto R.

### **3.2. Algoritmo de geração de colunas para resolver problemas gerais de planejamento florestal com restrições de adjacência**

WEINTRAUB (1994) propõe o algoritmo de geração de colunas para problemas gerais de planejamento florestal com restrições de adjacência.

A abordagem proposta considera uma formulação geral de programação linear para planejamento florestal tal como o FORPLAN, no qual são somadas restrições de adjacência. Resolver problemas de programação linear por geração de colunas é um procedimento usado para o caso onde o número de variáveis é muito grande. Em vez de declarar o problema explicitamente, são produzidas colunas para o problema mestre para resolver vários subproblemas. A solução ótima é encontrada quando o sub-problema não pode mais produzir nenhuma alternativa melhor do que as já produzidas no problema mestre. Uma aplicação de geração de colunas é o problema de cutting-stock (GILMORE e GOMORY, 1961). O procedimento é baseado no simplex revisado.

A idéia do método de geração de colunas consiste, como o próprio nome sugere, em ir gerando gradativamente as colunas da matriz A do PPLI. O algoritmo de geração de colunas também foi utilizado por CINTRA (2001) para resolver o problema de corte de estoque unidimensional e por RIBEIRO FILHO e LORENA (2000) onde fizeram uso combinado de Algoritmo Genético e geração de colunas para resolver de forma aproximada problemas de coloração de grafos.

Na abordagem proposta por WEINTRAUB são consideradas as seguintes condições:

1. O problema tem múltiplos períodos e um número significativo de restrições para serem satisfeitas, como exigência de madeira, restrições de orçamento, instalações e recreação para ser providenciado, controle de abastecimento de água e restrições de adjacência.
2. O problema é representado por grandes áreas de floresta, a qual pode incluir milhares de unidades de corte.

Para o problema modelo foram definidas zonas (que podem ser a floresta toda), onde cada zona é composta por áreas de administração homogêneas e as áreas são divididas em unidades de corte, sobre as quais recaem as restrições de adjacência. Para cada área as alternativas administrativas são definidas conforme o modelo I (JOHNSON e SCHEURMAN, 1977).

A figura a seguir mostra a representação de zona, área e unidade de corte utilizada na abordagem proposta por WEINTRAUB.

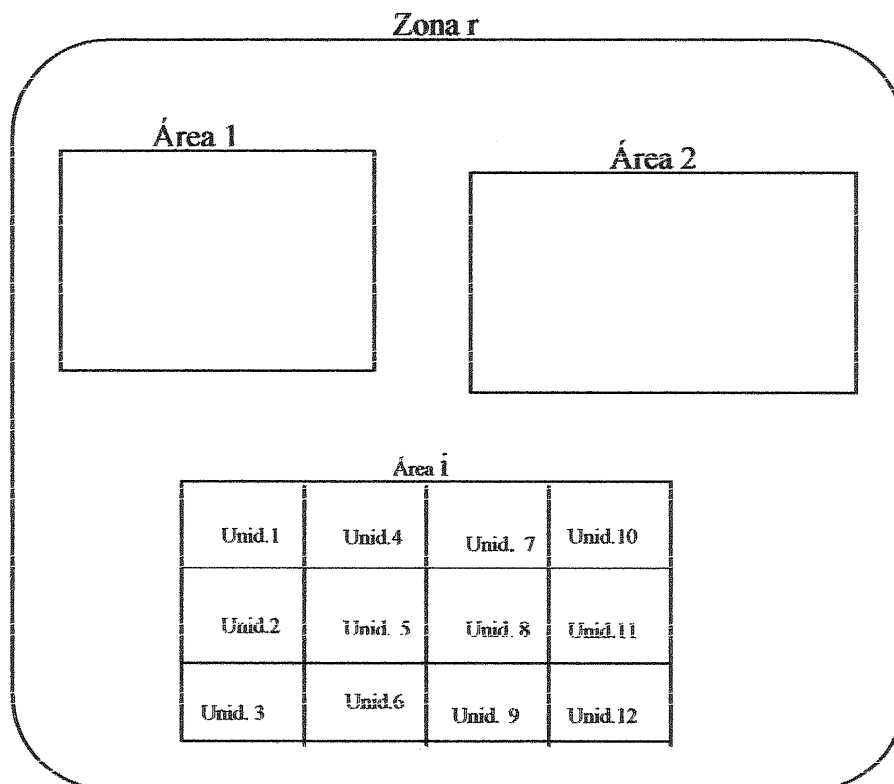


Figura 3.2.1: Representação esquemática de zona, área e unidade.

A alternativa de manejo para cada área significa um conjunto de decisões sobre o tempo para cada unidade de uma área. As alternativas devem ser tal que nenhum corte de madeira seja executado em unidades vizinhas em um mesmo período. Assim:

$$X_{ij} = \begin{cases} 1 & \text{se a área } i \text{ é manejada com a alternativa } j \\ 0 & \text{caso contrário} \end{cases}$$

Para o modelo foi considerado em cada período uma restrição de produção mínima de madeira e para cada zona foi acrescentada uma restrição de adjacência de tal forma que lá deva existir uma área mínima de madeira madura.

Para a formulação do modelo foram utilizadas os seguintes dados:

$A_{ijt}$  = produção total de madeira para a área  $i$  no período  $t$  se o manejo  $j$  é selecionado.

$F_t$  = produção mínima de madeira exigida para o período  $t$ .

$L_{ijt}$  = acres de madeira em pé na área  $i$  no período  $t$  seguindo o manejo  $j$ .

$C_{ij}$  = benefício presente líquido pelo manejo da área  $i$  com a alternativa  $j$ .

$D_{rt}$  = número mínimo de acres de madeira madura em pé, requisitados para a zona  $r$  no período  $t$ .

O problema mestre (M) foi formulado da seguinte maneira:

$$\text{Max } Z = \sum_i \sum_j c_{ij} x_{ij}$$

(maximizar o valor presente líquido)

Restrições:

$$(3.2.1) \sum_j x_{ij} = 1 \quad \forall i$$

$$(3.2.2) \sum_i \sum_j A_{ijt} x_{ij} \geq F_t \quad \forall t$$

$$(3.2.3) \sum_{i \in Z_r} \sum_j L_{ijt} x_{ij} \geq D_{rt} \quad \forall t, r$$

$$(3.2.4) x_{ij} \in \{0,1\} \quad \forall i, j$$

A restrição (3.2.1) representa que cada área é manejada por apenas uma alternativa  $j$ . Observando-se que as variáveis devem ser binárias para que as restrições de adjacência sejam satisfeitas.

A restrição (3.2.2) é a produção mínima de madeira requerida no período  $t$ .

Na restrição (3.2.3)  $Z_r$  é o conjunto de áreas da zona  $r$ , e ela garante para cada zona e período um número mínimo de acres com madeira madura em pé.

O problema  $M$  é um problema de programação inteira com um número grande de variáveis. Então a restrição de integralidade foi relaxada para  $0 \leq x_{ij} \leq 1$  (3.2.5) e este problema foi denotado  $M'$ . O problema  $M'$  foi resolvido utilizando-se a geração de colunas e em seguida a técnica Heurística Ávida (greedy heuristic) foi usada para encontrar uma solução inteira para  $M$ , a partir de uma solução contínua de  $M'$ .

Com o uso da Heurística Ávida para gerar soluções, a função objetivo alcançou um valor entre 96% e 99,9% do valor obtido na solução final. O tempo necessário para determinar as soluções candidatas foi de aproximadamente 1 segundo por área.

Na realização dos testes efetuados por WEINTRAUB a redução no valor da função objetivo com solução inteira a partir da solução contínua de  $M'$  teve seu pior caso em 1% e em média a redução foi de 0,3%. O que é mais significativo é que quando as dimensões do problema crescem, o erro tende a decrescer. Isto é explicado devido ao fato de que com um número maior de áreas, torna-se mais flexível à designação de alternativas que assegurem o valor do lado direito da desigualdade.

Outros pesquisadores também enfocaram o uso de restrições de adjacência no planejamento florestal, tais como:

HOGANSON e BORGES (1998) e BORGES et al. (1999) utilizaram programação dinâmica para focar o problema de adjacência em problemas de planejamento de grandes colheitas. Para vencer o

problema típico do tamanho do modelo a estratégia proposta foi unir formulações de programação dinâmica para coincidir subproblemas retirados do problema inicial. Estes subproblemas são resolvidos seqüencialmente. A estratégia utilizada encontrou boas soluções para três casos simples utilizados para testes.

MCDILL e BRAZE (2000), compararam três tipos de formulações para restrições de adjacência utilizando quatro distribuições de classes de idades para a floresta. O modelo de planejamento florestal foi formulado seguindo o modelo I e além de restrições de adjacência considerou restrições de fluxo e uma idade mínima média para o fim da floresta. Neste estudo observou-se que a distribuição por classes de idade é um fator importante que determina o tempo de solução do problema de planejamento florestal com restrições de adjacência.

BOSTON e BETTINGER (1999), compararam três técnicas heurísticas comumente utilizadas para resolver problemas de planejamento de colheitas com restrições de adjacência: Programação Inteira Monte Carlo, Simulated Annealing e Busca Tabu. Para as simulações neste trabalho foram utilizados quatro problemas, os quais tinham entre 3000 e 5000 variáveis binárias e além da solução heurística foi encontrada a solução dos problemas por meio técnicas de programação inteira para fazer a avaliação das soluções encontradas. A Busca Tabu localizou a melhor solução nos quatro problemas de planejamento analisados, enquanto a Programação Inteira Monte Carlo teve o menor valor para a função objetivo nos testes realizados.

No capítulo seguinte trataremos da formulação das restrições de adjacência proposta neste trabalho e também o modelo florestal considerado.

## CAPÍTULO IV

### 4. MATERIAL E MÉTODOS

O modelo de planejamento florestal considerado se refere a uma empresa fictícia que produz madeira para venda. Esta empresa deve atender os pedidos de seus clientes por sortimento de produtos, classificados em processo, laminação e serraria, sendo que seu objetivo é maximizar sua receita.

A empresa possui a prognose de crescimento da suas florestas, tendo a seu dispor os dados de volumes de madeira para cada produto disponíveis em cada talhão e cada período. O diâmetro na ponta fina com casca, geralmente utilizado, é de 8cm para madeira para processo, 18cm para serraria e acima de 25 cm para laminação.

A empresa também considera restrições de corte, ou seja, eia corta um talhão, em sua totalidade, uma única vez no horizonte de planejamento e além dessas restrições são consideradas restrições de adjacência que serão impostas a talhões vizinhos para que os mesmos não sejam cortados em um mesmo período e em períodos consecutivos.

O objetivo desta consideração é devida a requisitos ambientais como, por exemplo, conservar o habitat natural e evitar que grandes áreas de terra sejam devastadas em um mesmo local, evitando erosão, entre outros prejuízos ao meio ambiente.

Além das restrições de adjacência que são impostas a talhões vizinhos elas também podem considerar talhões que estejam distantes uns dos outros, neste caso talhões muito distantes não seriam colhidos em um mesmo período, evitando assim um aumento nos custos com a

manutenção de estradas e o transporte, custos estes os mais elevados em todo o processo de planejamento da floresta.

A determinação dos talhões distantes dependerá das necessidades da empresa. Poderão ser considerados distantes os talhões que estiverem a uma distância pré-determinada ou ainda, talhões que necessitem a construção ou manutenção de estradas para seu acesso. Esta necessidade poderá inviabilizar a colheita de tais talhões no mesmo período, devido a este fato acarretar um aumento considerável nos custos.

Em seguida mostraremos, com um exemplo, a formulação matemática das restrições de adjacência. A primeira formulação será para impedir que talhões vizinhos sejam colhidos em um mesmo período, esta formulação será denominada como 1º caso. Em seguida far-se-á formulação para impedir a colheita de talhões vizinhos em períodos consecutivos, denominada de 2º caso. A última formulação será para a consideração de talhões vizinhos e talhões distantes em um mesmo período, 3º caso.

#### 4.1 Formulação das restrições de adjacência para evitar a colheita de talhões vizinhos em um mesmo período

Para a formulação das restrições de adjacência consideraremos um horizonte de planejamento de 10 períodos e uma área com a seguinte distribuição de talhões:

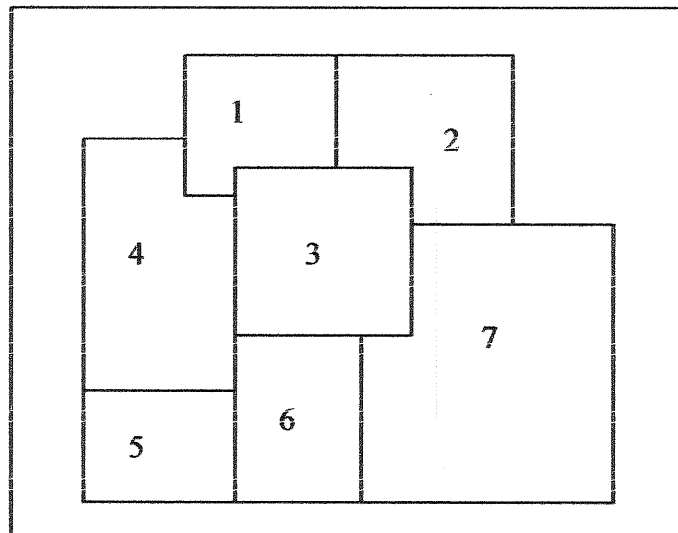


Figura 4.1.1 Distribuição dos talhões

A variável de decisão envolvida neste modelo é:

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é cortado no período } k \\ 0 & \text{caso contrário} \end{cases}$$

Consideremos a imposição de que talhões vizinhos não devem ser colhidos em um mesmo período. Tomemos por exemplo, o talhão 1, caso este seja selecionado para corte no período 3, os seus vizinhos, talhões 2, 3 e 4 não poderão ser selecionados para corte neste período.

Para a formulação das restrições de adjacência precisa-se identificar os vizinhos de cada talhão e registrá-los em uma matriz de vizinhos denominada  $A$ , composta por elementos zero ou um. O elemento será zero quando os talhões não forem vizinhos, assim:

$$a_{ij} = \begin{cases} 1 & \text{se o talhão } i \text{ é vizinho do talhão } j \\ 0 & \text{caso contrário} \end{cases}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Na matriz A, a 1ª linha representa os vizinhos do talhão 1 e assim sucessivamente até a linha 7 que são os vizinhos do talhão 7.

Consideremos, por exemplo, o talhão 3 em um determinado período k, sendo o 3 selecionado, os talhões 1, 2, 4, 6 e 7 não podem ser selecionados por serem vizinhos de 3. Assim poderíamos escrever:

$$x_{3k} + x_{1k} + x_{2k} + x_{4k} + x_{6k} + x_{7k} \leq 1 \quad (4.1.1)$$

Esta restrição permite que somente um dos talhões considerados seja selecionado.

Por outro lado, se 3 não é selecionado existem combinações possíveis entre 1, 2, 4, 6 e 7 que devem ser permitidas pela restrição. Para permitir a combinação de colheita com estes talhões necessitamos de um coeficiente diferente de um para o talhão 3. Na formulação considerada este coeficiente é determinado pelo número de talhões envolvidos na restrição e este mesmo valor deverá ser considerado no lado direito da desigualdade.

Desta forma, temos:

$$6x_{3k} + x_{1k} + x_{2k} + x_{4k} + x_{6k} + x_{7k} \leq 6 \quad (4.1.2)$$

O coeficiente de  $x_{3k}$  na equação (4.1.2) garante que se o talhão 3 for selecionado para a colheita no período k os demais talhões presentes na

inequação não o serão, pois o valor do coeficiente de  $x_{3k}$  é igual ao valor do lado direito da inequação. Também observamos que esta restrição permite qualquer combinação com os vizinhos do talhão 3, tendo em vista que seus coeficientes são 1, a sua soma não ultrapassará o valor do lado direito da desigualdade, uma vez que ela é obtida pelo número de vizinhos de 3 mais 1, que representa o próprio talhão 3.

Generalizando para um talhão  $i$  qualquer e em um período  $k$ , temos:

$$\left( \sum_{j=1}^n a_{ij} \cdot x_{jk} \right) + \sum_{j=1}^n a_{ij} \cdot x_{jk} \leq \sum_{j=1}^n a_{ij} \quad (4.1.3) \quad \text{ou}$$

$$\sum_{j=1}^n a_{ij} \cdot x_{jk} \leq \sum_{j=1}^n (a_{ij} (1 - x_{jk})) \quad (4.1.4)$$

$$i = 1, 2, \dots, n$$

$$k = 1, 2, \dots, t$$

A equação (4.1.4) é a forma geral para restrições de adjacência impostas a talhões vizinhos em um mesmo período, onde:

$i, j$  são os índices referentes aos talhões

$n$  é o número de talhões

$a_{ij}$  indica se o talhão  $i$  é vizinho do talhão  $j$

$$x_{ik} (x_{jk}) = \begin{cases} 1 & \text{se o talhão } i \text{ (} j \text{) é selecionado no período } k \\ 0 & \text{caso contrário.} \end{cases}$$

## 4.2 Formulação das restrições de adjacência para impedir a colheita de talhões vizinhos em períodos consecutivos

Para a formulação das restrições de adjacência para períodos consecutivos consideraremos o mesmo exemplo anterior.

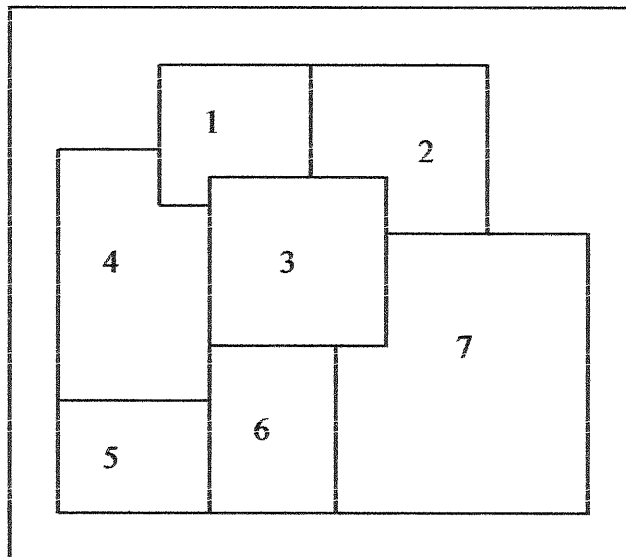


Figura 4.2.1 Distribuição dos talhões

As variáveis de decisão e a matriz A de vizinhos são as mesmas definidas anteriormente.

Consideremos, novamente, o talhão 3. Se este é seleccionado no período k, as restrições de adjacência não deverão permitir que os talhões 1, 2, 4, 6 e 7 sejam colhidos no período k e no período k+1. Desta forma a equação que garante que os talhões vizinhos de 3 não sejam seleccionados junto com ele no mesmo período é:

$$6x_{3k} + x_{1k} + x_{2k} + x_{4k} + x_{6k} + x_{7k} \leq 6 \quad (4.2.1)$$

Para que estes talhões também não sejam seleccionados no período seguinte acrescentaremos uma restrição de adjacência que considerará o talhão 3 no período k e seus vizinhos no período k+1, assim:

$$6x_{3k} + x_{1(k+1)} + x_{2(k+1)} + x_{4(k+1)} + x_{6(k+1)} + x_{7(k+1)} \leq 6 \quad (4.2.2)$$

Generalizando estas equações para um talhão  $i$  e um período  $k$  temos:

$$\sum_{j=1}^n a_{ij} \cdot x_{jk} \leq \sum_{j=1}^n (a_{ij} (1 - x_{ik})) \quad (4.2.3)$$

$$i = 1, 2, \dots, n$$

$$k = 1, 2, \dots, t$$

$$\sum_{j=1}^n a_{ij} \cdot x_{j(k+1)} \leq \sum_{j=1}^n (a_{ij} (1 - x_{ik})) \quad (4.2.4)$$

$$i = 1, 2, \dots, n$$

$$k = 1, 2, \dots, t-1$$

As equações (4.2.3) e (4.2.4) são as equações gerais para que talhões vizinhos não sejam colhidos em um mesmo período e no período consecutivo. Observamos que na equação (4.2.4) o valor de  $k$  não pode variar até o último período do horizonte de planejamento porque os talhões  $j$ , vizinhos do talhão  $i$ , são considerados no período  $(k+1)$ , assim  $k$  varia até o penúltimo período do horizonte de planejamento.

#### 4.3 Formulação das restrições de adjacência para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período

Consideremos novamente a mesma definição dos talhões dos exemplos anteriores.

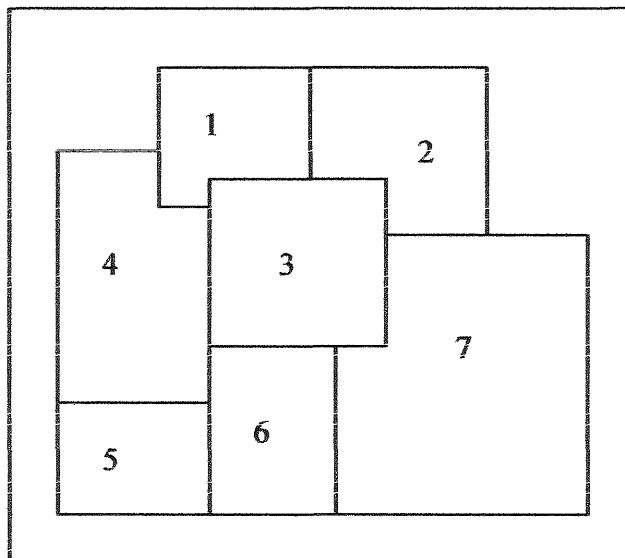


Figura 4.3.1 Distribuição dos talhões

Para este caso, suponhamos que a empresa não deseje que os talhões 1 e 7 sejam colhidos no mesmo período, devido a distância entre eles e também porque eles necessitam de manutenção de estradas para seu acesso. Neste caso, a empresa deseja que em um período seja arrumada apenas uma estrada de acesso aos talhões. A empresa também não deseja colher os talhões 2 e 5 no mesmo período devido a distância entre eles.

Para a solução deste problema a variável de decisão é dada por:

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é colhido no período } k \\ 0 & \text{caso contrário} \end{cases}$$

Para a construção da matriz de vizinhos  $A$ , devemos registrar os talhões vizinhos e os talhões distantes do talhão  $i$  considerado com o número um e os demais com zero, desta forma:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Observamos que na primeira linha temos registrados com o número um os talhões 1, 2, 3, 4 e 7. Os talhões 2, 3, e 4 são vizinhos dos talhão 1 e o talhão 7 é o talhão considerado distante de 1.

A restrição que impede que estes talhões sejam seleccionados simultaneamente em um período  $k$  é:

$$5x_{1k} + x_{2k} + x_{3k} + x_{4k} + x_{7k} \leq 5 \quad (4.3.1)$$

Generalizando esta equação para um talhão  $i$  e um período  $k$ , temos:

$$\sum_{j=1}^n a_{ij} x_{jk} \leq \sum_{j=1}^n (a_{ij} (1 - x_{ik})) \quad (4.3.2)$$

$$i = 1, 2, \dots, n$$

$$k = 1, 2, \dots, t$$

$$a_{ij} = \begin{cases} 1 & \text{se o talhão } i \text{ é vizinho ou distante do talhão } j \\ 0 & \text{caso contrário} \end{cases}$$

A equação (4.3.2) garante que os vizinhos do talhão  $i$  e os talhões designados como distantes dele não sejam seleccionados simultaneamente em um mesmo período  $k$ .

A equação (4.3.2) é idêntica a equação (4.1.4) que é utilizada para impedir a colheita de talhões vizinhos em um mesmo período, a única diferença entre as duas formulações está na forma de registro dos

elementos da matriz A. Na equação (4.1.4) os elementos  $a_{ij}$  indicam os talhões vizinhos do talhão i, e na equação (4.3.2) os elementos  $a_{ij}$  indicam os talhões que são vizinhos do talhão i e também os talhões que são considerados distantes do talhão i.

A seguir faremos a formulação da função objetivo e das demais restrições do modelo.

#### 4.4 Formulação do modelo completo

O modelo apresentado a seguir é um modelo de programação linear inteira binária que procura maximizar a receita de uma empresa florestal considerando as restrições de adjacência, restrições de corte e restrições de demanda dos produtos para processo, serraria e laminado.

##### 4.4.1 Função Objetivo

A função objetivo considerada para o problema foi maximizar a receita bruta ao longo do horizonte de planejamento. Assim:

$$\text{F.O. Max: } \sum_{i=1}^n \sum_{k=1}^t r_{ik} \cdot x_{ik} \text{ onde:}$$

$r_{ik}$  – é a receita obtida com a colheita do talhão i no período k.

$$x_{ik} = \begin{cases} 1 & \text{se o talhão i é cortado no período k} \\ 0 & \text{caso contrário} \end{cases}$$

##### 4.4.2 Restrições:

Cada restrição considerada neste modelo será descrita e apresentada matematicamente por meio de inequações lineares.

#### 4.4.2.1 Restrições de corte:

Estas restrições garantirão que os talhões sejam cortados uma única vez durante o horizonte de planejamento.

Representando por  $i$  o talhão,  $k$  o período,  $n$  o número total de talhões,  $t$  o total de períodos no horizonte de planejamento e  $x_{ik}$  como definido anteriormente, temos:

$$x_{i1} + x_{i2} + \dots + x_{it} \leq 1 \quad \text{ou}$$

$$\sum_{k=1}^t x_{ik} \leq 1 \quad \text{para } i=1,2,\dots,n$$

#### 4.4.2.2 Restrições de demanda:

Os produtos considerados neste modelo são: processo, serraria e laminado. Para cada um deles a empresa deve cumprir a demanda encomendada pelos seus clientes. Dessa forma, para cada período, cada um dos produtos terá um demanda fixada. Assim podemos apresentar a restrição de demanda da seguinte forma:

$$\sum_{i=1}^n V_{ikp} \cdot x_{ik} \geq D_{kp} \quad \begin{array}{l} k=1,2,\dots,t \\ p=1,2,3. \end{array}$$

Onde:

$p$  – índices que representam os produtos

$V_{ikp}$  – volume total disponível do produto  $p$ , no talhão  $i$  para o período  $k$

$D_{kp}$  – demanda necessária do produto  $p$  no período  $k$

Esta restrição garante que a demanda mínima seja atendida para cada período do horizonte de planejamento.

De acordo com estes itens, o modelo a ser considerado é o seguinte:

$$\text{F.O. Max: } \sum_{i=1}^n \sum_{k=1}^t r_{ik} \cdot x_{ik}$$

s.a.

$$(I) \sum_{k=1}^t x_{ik} \leq 1 \quad \text{para } i=1,2,\dots,n$$

$$(II) \sum_{i=1}^n x_{ik} \cdot v_{ikp} \geq D_{kp} \\ k=1,2,\dots,t \quad p=1,2,3.$$

$$(III) \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \cdot x_{jk} \leq \sum_{j=1}^n (a_{ij} (1 - x_{ik})) \\ i=1,2,\dots,n \quad k=1,2,\dots,t$$

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é cortado no período } k \\ 0 & \text{caso contrário} \end{cases}$$

Este modelo será utilizado para a consideração das restrições de adjacência para impedir a colheita de talhões vizinhos em um mesmo período e também para a consideração de restrições de adjacência que impeçam a colheita de talhões vizinhos e talhões distantes em um mesmo período. Observando-se que a diferença na formulação da restrição (III) encontra-se na forma de registro dos elementos da matriz A.

Para utilizar as restrições de adjacência para impedir a colheita de talhões vizinhos em períodos consecutivos acrescentamos a restrição a seguir no modelo:

$$(IV) \sum_{\substack{j=1 \\ j \neq i}}^{\bar{n}} x_{j(k+1)} \cdot a_{ij} \leq \sum_{j=1}^{\bar{n}} (a_{ij}(1 - x_{ik}))$$

$$i = 1, 2, \dots, n \quad k = 1, 2, \dots, (t - 1)$$

A seguir consideraremos alguns exemplos onde as restrições de adjacência serão impostas para impedir a colheita de talhões vizinhos em um mesmo período (1º caso), para impedir a colheita de talhões vizinhos em períodos consecutivos (2º caso) e também aplicaremos as restrições de adjacência para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período (3º caso), juntamente com suas soluções obtidas com o software LINGO. No anexo 1 serão descritos alguns comandos básicos e a sintaxe do LINGO para sua utilização na resolução de problemas de programação linear.

#### 4.5 Exemplo de aplicação das restrições de adjacência

Consideremos a seguinte distribuição de talhões em uma floresta:

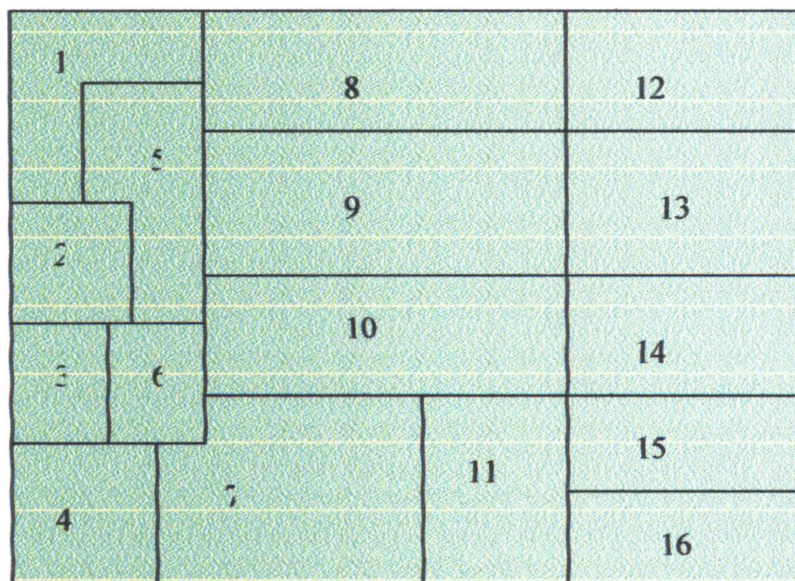


Figura 4.5.1: Distribuição dos 16 talhões

Formularemos o modelo considerando os 16 talhões representados na figura 5.4, acima, com horizonte de planejamento de 10 períodos procurando a maximização das receitas, utilizando restrições de corte, restrições de demanda pelos produtos laminação (produto 1), serraria (produto 2) e processo (produto 3). Também consideraremos restrições de adjacência que impeçam a colheita de talhões vizinhos em um mesmo período e em seguida em períodos consecutivos, também em um terceiro momento, utilizaremos as restrições de adjacência para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período. Para isto utilizaremos os seguintes dados: receita anual por talhão, volume de cada produto por talhão e por período e também as demandas desses produtos em cada período. As tabelas de receitas anuais e de volumes são apresentadas no anexo 2. Estes dados serão utilizados nas três formulações para que se possa comparar seus resultados.

A tabela 4.5.1 contém a demanda requisitada para cada produto em cada período.

Período	DEMANDAS (1000 m <sup>3</sup> )		
	Produto 1	Produto 2	Produto 3
1	5	40	60
2	15	40	60
3	15	40	60
4	20	60	90
5	25	60	90
6	30	60	90
7	30	60	90
8	50	100	120
9	65	100	150
10	95	150	200

Tabela 4.5.1: Demanda dos produtos 1, 2 e 3 para cada período horizonte de planejamento.

A matriz a seguir registra os vizinhos de cada talhão, se o elemento  $a_{ij}$  da matriz A for 1 significa que o talhão i é vizinho do talhão j, caso seja 0 significa que i não é vizinho de j.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Com estes dados podemos formular o problema.

$$\text{F.O. Max: } \sum_{i=1}^{16} \sum_{k=1}^{10} r_{ik} \cdot x_{ik} \text{ onde:}$$

s.a.

$$(I) \sum_{k=1}^{10} x_{ik} \leq 1 \quad \text{para } i=1,2,\dots,16$$

$$(II) \sum_{i=1}^{16} v_{ikp} \cdot x_{ik} \geq D_{kp} \quad k=1,2,\dots,10 \\ p=1,2,3.$$

$$(III) \sum_{\substack{j=1 \\ j \neq i}}^{16} x_{jk} \cdot a_{ij} \leq \sum_{j=1}^{16} (a_{ij}(1-x_{ik}))$$

$$i=1,2,\dots,16 \quad k=1,2,\dots,10$$

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é cortado no período } k \\ 0 & \text{caso contrário} \end{cases}$$

$i, j$  – são os índices referentes aos talhões (1 a 16)

$r_{ik}$  – receita propiciada pela colheita do talhão  $i$  no período  $k$

$v_{ikp}$  – volume de madeira do produto  $p$  no período  $k$  no talhão  $i$

$D_{kp}$  – demanda do produto  $p$  no período  $k$

$$a_{ij} = \begin{cases} 1 & \text{se o talhão } i \text{ é vizinho do talhão } j \\ 0 & \text{caso contrário} \end{cases}$$

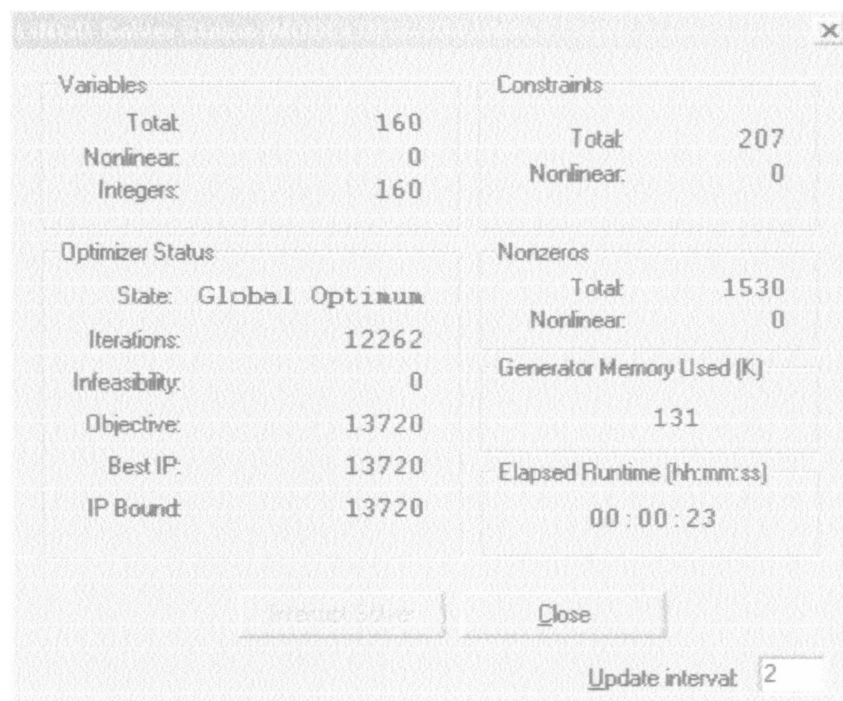
A formulação do problema utilizando a sintaxe do LINGO, bem como a definição dos conjuntos e dos dados utilizados no problema para os três casos estudados serão apresentados no anexo 3.

A função objetivo do problema procura maximizar a receita da floresta.

O primeiro grupo de restrições são definidas como restrições de corte e garantem que cada talhão seja selecionado para corte em um único período no horizonte de planejamento.

O segundo grupo de restrições referem-se às restrições de demanda e asseguram o requerimento mínimo de madeira para cada produto por período no horizonte de planejamento. No terceiro grupo de restrições encontramos as restrições de adjacência que impedirão a colheita de talhões vizinhos em um mesmo período.

Processado o problema a janela *solver status* do LINGO mostra todos os elementos do problema:



Variables		Constraints	
Total	160	Total	207
Nonlinear	0	Nonlinear	0
Integers	160		
Optimizer Status		Nonzeros	
State	Global Optimum	Total	1530
Iterations	12262	Nonlinear	0
Infeasibility	0	Generator Memory Used (K)	131
Objective	13720	Elapsed Runtime (hh:mm:ss)	00:00:23
Best IP	13720		
IP Bound	13720		
		Update interval	2

Figura 4.5.2 : Janela do LINGO, *solver status*

O problema proposto é composto por 160 variáveis binárias e 207 restrições lineares, sendo que 16 são restrições de corte, 30 são restrições de demanda, 160 são restrições de adjacência e 1 restrição para a definição das variáveis como binárias.

A figura 4.5.3 mostra a solução do problema apresentada pelo LINGO:

```

Command Window
X( 1, 1)      1.000000      -250.0000
X( 2, 7)      1.000000      -294.0000
X( 3, 10)     1.000000      -440.0000
X( 4, 4)       1.000000      -372.0000
X( 5, 3)       1.000000      -316.0000
X( 6, 7)       1.000000      -220.5000
X( 7, 10)     1.000000      -2250.0000
X( 8, 9)       1.000000      -1507.5000
X( 9, 10)     1.000000      -2250.0000
X( 10, 2)     1.000000      -450.0000
X( 11, 9)     1.000000      -1206.0000
X( 12, 10)   1.000000      -1012.5000
X( 13, 8)     1.000000      -1068.0000
X( 14, 10)   1.000000      -1012.5000
X( 15, 6)     1.000000      -571.5000
X( 16, 5)     1.000000      -499.5000
  
```

Figura 4.5.3: solução do problema

A solução apresentada na figura 4.5.3 nos fornece os talhões que devem ser colhidos em cada período. Este resultado é resumido na tabela 4.5.2 a seguir:

Período	talhões
1	1
2	10
3	5
4	4
5	16
6	15
7	2 – 6
8	13
9	8 – 11
10	3 – 7 - 9 - 12 - 14

Tabela 4.5.2: Solução do problema com 16 talhões e 10 períodos.

Observando a tabela verificamos que as condições impostas pelas restrições de adjacência foram atendidas, nenhum talhão vizinho foi colhido em um mesmo período. O valor encontrado para a função objetivo é  $Z = \$ 13.720$ , sendo o problema resolvido em 23 segundos com 12.262 iterações.

Também foram efetuados testes para estes mesmos talhões onde as restrições de adjacência impedem que talhões vizinhos sejam cortados em períodos consecutivos, assim se o talhão 1 é colhido no período 1 os seus vizinhos não podem ser colhidos nos períodos 1 e 2. A formulação matemática para este caso é a seguinte:

$$\text{F.O. Max: } \sum_{i=1}^{16} \sum_{k=1}^{10} r_{ik} \cdot x_{ik} \text{ onde:}$$

s.a.

$$(I) \sum_{k=1}^{10} x_{ik} \leq 1 \quad \text{para } i=1,2,\dots,16$$

$$(II) \sum_{i=1}^{16} v_{ikp} \cdot x_{ik} \geq D_{kp} \quad \begin{array}{l} k = 1,2,\dots,10 \\ p = 1,2,3. \end{array}$$

$$(III) \sum_{\substack{j=1 \\ j \neq i}}^{16} x_{jk} \cdot a_{ij} \leq \sum_{j=1}^{16} (a_{ij}(1 - x_{ik}))$$

$$i = 1,2, \dots, 16 \quad k = 1, 2, \dots, 10$$

$$(IV) \sum_{\substack{j=1 \\ j \neq i}}^{16} x_{j(k+1)} \cdot a_{ij} \leq \sum_{j=1}^{16} (a_{ij}(1 - x_{ik}))$$

$$i = 1, 2, \dots, 16 \quad k = 1, 2, \dots, 9$$

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é cortado no período } k \\ 0 & \text{caso contrário} \end{cases}$$

$i, j$  – são os índices referentes aos talhões (1 a 16)

$r_{ik}$  – receita propiciada pela colheita do talhão  $i$  no período  $k$

$V_{ikp}$  – volume de madeira do produto  $p$  no período  $k$  no talhão  $i$

$D_{kp}$  – demanda do produto  $p$  no período  $k$

$$a_{ij} = \begin{cases} 1 & \text{se o talhão } i \text{ é vizinho do talhão } j \\ 0 & \text{caso contrário} \end{cases}$$

A função objetivo e os três primeiros grupos de restrições são os mesmos já descritos anteriormente. O segundo grupo de restrições de adjacência garantem a condição de que talhões vizinhos não sejam colhidos em períodos consecutivos, uma vez que ela considera o talhão  $i$  no tempo  $k$  e os talhões  $j$ , vizinhos de  $i$ , no período  $(k+1)$ .

A solução obtida para este problema está resumida na tabela abaixo:

Período	talhões
1	13
2	10
3	4
4	5
5	15
6	8
7	2 - 6
8	11
9	1
10	3 - 7 - 9 - 12 - 14 - 16

Tabela 4.5.3: Solução do problema com 16 talhões e 10 períodos, com restrições de adjacência para períodos consecutivos.

O problema foi resolvido em 24 segundos com 8162 iterações e o valor da função objetivo encontrado foi \$13.455. Este problema contava com 160 variáveis binárias e 351 restrições, das quais 16 são restrições de corte, 30 são restrições de demanda, 160 são restrições de adjacência para impedir a colheita de talhões vizinhos em um mesmo período e 144 referem-se as restrições de adjacência que garantirão que talhões vizinhos não sejam selecionados em períodos consecutivos.

Outra situação avaliada diz respeito às restrições de adjacência serem aplicadas aos talhões vizinhos e talhões distantes, desta forma elas impedirão que talhões considerados distantes sejam colhidos em um mesmo período, procurando desta maneira minimizar o custo da colheita. Para alcançarmos este objetivo devemos registrar na matriz de vizinhos tanto os talhões vizinhos como os talhões distantes com número um. Considerando os mesmos dados dos problemas anteriores teremos a seguinte formulação matemática:

$$\text{F.O. Max: } \sum_{i=1}^{16} \sum_{k=1}^{10} r_{ik} \cdot x_{ik} \text{ onde:}$$

s.a.

$$(I) \sum_{k=1}^{10} x_{ik} \leq 1 \quad \text{para } i=1,2,\dots,16$$

$$(II) \sum_{i=1}^{16} v_{ikp} \cdot x_{ik} \geq D_{kp} \quad \begin{array}{l} k=1,2,\dots,10 \\ p=1,2,3. \end{array}$$

$$(III) \sum_{\substack{j=1 \\ j \neq i}}^{16} x_{jk} \cdot a_{ij} \leq \sum_{j=1}^{16} (a_{ij} (1 - x_{ik}))$$

$$i = 1,2, \dots, 16 \quad k = 1, 2, \dots, 10$$

$$x_{ik} = \begin{cases} 1 & \text{se o talhão } i \text{ é cortado no período } k \\ 0 & \text{caso contrário} \end{cases}$$

$i, j$  – são os índices referentes aos talhões (1 a 16)

$r_{ik}$  – receita propiciada pela colheita do talhão  $i$  no período  $k$

$V_{ikp}$  – volume de madeira do produto  $p$  no período  $k$  no talhão  $i$

$D_{kp}$  – demanda do produto  $p$  no período  $k$

$$a_{ij} = \begin{cases} 1 & \text{se o talhão } i \text{ é vizinho ou é distante do talhão } j \\ 0 & \text{caso contrário} \end{cases}$$

Nesta formulação todas as restrições são as mesmas já descritas no primeiro caso, a única diferença é a matriz  $A$  que contém os talhões vizinhos e os talhões distantes de cada talhão  $i$  registrados com número um na posição indicada.

A solução encontrada está registrada na tabela 4.5.4. Este modelo continha 160 variáveis binárias, 207 restrições, o valor encontrado para a função objetivo foi \$ 13.597,5 com 816.548 iterações em 23 minutos e 58 segundos.

Período	Talhões
1	1
2	10
3	4
4	5
5	15
6	12
7	14
8	6 – 11
9	2 – 9
10	3 – 7 – 8 – 13 – 16

Tabela 4.5.4: Solução do problema com 16 talhões e 10 períodos, com restrições de adjacência para talhões vizinhos e talhões distantes.

Para esta formulação observamos que todas as restrições foram atendidas, ou seja, nenhum talhão vizinho foi selecionado para corte no mesmo período e também nenhum dos talhões considerados como distantes foram selecionados para colheita no mesmo período.

No capítulo seguinte será feita a análise dos resultados obtidos em alguns testes efetuados nos quais foram variados o número de períodos no horizonte de planejamento e também o número de talhões que compõe a área florestal.

## CAPÍTULO V

### 5. ANÁLISE DOS RESULTADOS

Neste capítulo será feita a análise dos resultados obtidos nos testes realizados para avaliar a validade da formulação das restrições de adjacência proposta neste trabalho.

Os testes foram realizados com número diferentes de períodos no horizonte de planejamento e também diferentes números de talhões compondo a área florestal estudada. Em todas as situações foram avaliadas as três formulações propostas para as restrições de adjacência, ou seja, no 1º caso avaliou-se as restrições de adjacência quando impostas para impedir a colheita de talhões vizinhos em um mesmo período. No 2º caso as restrições de adjacência impediam a colheita de talhões vizinhos em períodos consecutivos e no 3º caso as restrições de adjacência não permitiam a colheita de talhões vizinhos e talhões distantes em um mesmo período.

O primeiro teste realizado foi com 16 talhões e 10 períodos no horizonte de planejamento.

No quadro 5.1 estão resumidos os resultados obtidos com 16 talhões e 10 períodos no horizonte de planejamento para a formulação das restrições de adjacência nos três casos estudados. Estes resultados referem-se às formulações exemplificadas no capítulo anterior.

<b>Caso estudado</b>	<b>Número de variáveis</b>	<b>Número de restrições</b>	<b>Valor da função objetivo</b>	<b>Tempo de resolução</b>	<b>Número de iterações</b>
1º caso	160	207	13720	23 seg	12.262
2º caso	160	351	13455	24 seg	8.162
3º caso	160	207	13597,5	23 min 58 seg	816.548

Quadro 5.1. Resultados dos testes com 16 talhões.

A primeira linha do quadro 5.1 refere-se ao resultado do problema no qual as restrições de adjacência são aplicadas para impedir a colheita de talhões vizinhos em um mesmo período.

A segunda linha do quadro contém os resultados da aplicação das restrições de adjacência impedindo a colheita de talhões vizinhos em períodos consecutivos e, a última linha traz os resultados do problema no qual as restrições de adjacência são aplicadas para impedir que talhões vizinhos e distantes sejam colhidos em um mesmo período.

Este mesmo problema com 16 talhões e 10 períodos no horizonte de planejamento foi resolvido sem a utilização de restrições de adjacência. O valor da função objetivo encontrada foi \$13.983,5 sendo necessários 9 minutos e 27 segundos para sua resolução.

Observando o valor da função objetivo no primeiro caso, \$13.720, onde as restrições de adjacência foram impostas para talhões vizinhos em um mesmo período, podemos notar que a redução no valor da função objetivo não foi muito grande, levando em consideração que sem as restrições de adjacência teríamos como solução ótima colher no período 10, por exemplo, os talhões 6, 7, 8, 9, 12 e 14 o que representa uma grande área devastada em um mesmo local.

Neste caso, a perda monetária no valor da função objetivo pode ser compensada pelo ganho ambiental propiciado pelo uso das restrições de adjacência.

O tempo gasto para a resolução do problema com as restrições de adjacência foi 23 segundos. Comparando-se com a resolução sem restrições de adjacência observamos uma redução considerável, embora, ambos sejam irrelevantes por estarmos tratando de um planejamento de longo prazo.

Nesta primeira formulação o objetivo foi alcançado, tendo em vista que nenhum talhão vizinho foi selecionado para corte no mesmo período.

Na segunda formulação houve uma redução na função objetivo para \$13.455. Nesta formulação foram utilizadas as restrições de adjacência para impedir a colheita de talhões vizinhos em períodos consecutivos.

Podemos considerar o fato de estarmos trabalhando com um número pequeno de talhões um dos fatores que influenciou nesta redução de aproximadamente 2% em relação ao valor da função objetivo no primeiro caso.

No terceiro caso, onde as restrições de adjacência foram impostas a talhões vizinhos e distantes, o valor da função objetivo encontrada foi \$13.597,5. Neste caso, a redução não chegou a 1% do valor obtido no primeiro caso. Pode-se verificar também, que na solução encontrada os talhões selecionados no mesmo período não são vizinhos e também não estão muito dispersos uns dos outros, facilitando assim o deslocamento de maquinários e funcionários até os locais de colheita.

Os demais testes realizados foram considerando 25 talhões e 15 períodos no horizonte de planejamento, 32 talhões e 15 períodos no

horizonte de planejamento e 50 talhões com 30 períodos no horizonte de planejamento.

A distribuição dos talhões, o modelo formulado para as simulações e a solução de cada uma encontram-se no anexo 4.

Os resultados destes testes estão registrados nos quadros a seguir:

<b>Caso estudado</b>	<b>Número de variáveis</b>	<b>Número de restrições</b>	<b>Valor da função objetivo</b>	<b>Tempo de resolução</b>	<b>Número de iterações</b>
1º caso	375	446	818,9	7 seg	1.563
2º caso	375	796	816,8	18 seg	3.216
3º caso	375	446	807,2	1min. 38 seg	21.743

Quadro 5.2: Resultado das simulações para 25 talhões com 15 anos no horizonte de planejamento.

<b>Caso estudado</b>	<b>Número de variáveis</b>	<b>Número de restrições</b>	<b>Valor da função objetivo</b>	<b>Tempo de resolução</b>	<b>Número de iterações</b>
1º caso	480	558	1083,3	4 seg	345
2º caso	480	1006	1064,5	59 min 45 seg	399.804
3º caso	480	558	1067,9	16 min 5 seg	151.133

Quadro 5.3: Resultado das simulações para 32 talhões com 15 anos no horizonte de planejamento.

<b>Caso estudado</b>	<b>Número de variáveis</b>	<b>Número de restrições</b>	<b>Valor da função objetivo</b>	<b>Tempo de resolução</b>	<b>Número de iterações</b>
1º caso	1500	1641	1647,1	6 seg	548
2º caso	1500	3091	1646	13 seg	1.613
3º caso	1500	1641	1618,9	10 min 45 seg	41.562

Quadro 5.4: Resultado das simulações para 50 talhões com 30 anos no horizonte de planejamento.

Em todos os testes os resultados mostraram-se satisfatórios, uma vez que em cada proposta as restrições impostas foram atendidas.

Pode-se verificar que o valor da função objetivo não teve uma redução muito grande comparando-se a primeira formulação com as formulações para anos consecutivos e para talhões vizinhos e talhões distantes. A maior redução na função objetivo não chegou a 2% do valor da função objetivo no modelo considerando restrições de adjacência para o mesmo período.

Em termos de tempo de resolução observamos o maior gasto na simulação com 32 talhões e 15 períodos para restrições de adjacência impostas para talhões vizinhos em períodos consecutivos onde se obteve a solução em 59 minutos e 45 segundos.

Um fato que pode ter influenciado no tempo de resolução é a distribuição dos talhões, pois neste problema foi considerada uma região bem recortada, onde mais ou menos 50% dos talhões tem quatro ou mais vizinhos, tendo talhões com até 6 vizinhos, enquanto nos demais casos os talhões tem no máximo quatro vizinhos.

Para observarmos a ação das restrições de adjacência sobre talhões vizinhos e distantes usaremos a solução dada para o problema com 50 talhões e 30 anos no horizonte de planejamento.

As figuras a seguir mostram os talhões que devem ser cortados nos anos 10, 20 e 30. Anos estes em que são selecionados mais talhões simultaneamente para corte. Os demais talhões da unidade florestal são colhidos nos demais períodos do horizonte de planejamento.




1	11	21	31	41
2	12	22	32	42
3	13	23	33	43
4	14	24	34	44
5	15	25	35	45
6	16	26	36	46
7	17	27	37	47
8	18	28	38	48
9	19	29	39	49
10	20	30	40	50

Figura 5.1: Talhões que devem ser colhidos nos anos 10, 20 e 30, com restrições de adjacência impostas para talhões vizinhos em um mesmo período.

1	11	21	31	41
2	12	22	32	42
3	13	23	33	43
4	14	24	34	44
5	15	25	35	45
6	16	26	36	46
7	17	27	37	47
8	18	28	38	48
9	19	29	39	49
10	20	30	40	50

Figura 5.2: Talhões que devem ser colhidos nos anos 10, 20 e 30, com restrições de adjacência impostas a talhões vizinhos e talhões distantes.

Para as figuras 5.1 e 5.2 utilizamos a seguinte legenda:

-  Talhões selecionados para corte no ano 10
-  Talhões selecionados para corte no ano 20
-  Talhões selecionados para corte no ano 30

Comparando os resultados das duas figuras observamos que ao utilizarmos as restrições de adjacência para talhões vizinhos e distantes obtivemos uma concentração maior dos talhões em um mesmo período sem que talhões vizinhos fossem selecionados. Na figura 5.1 temos talhões distribuídos em toda a floresta para a colheita no ano 30 enquanto no segundo caso os talhões selecionados para o ano 30 estão distribuídos em duas faixas. Na primeira situação observamos um número maior de talhões selecionados no ano 30, já na segunda situação,

houve uma distribuição destes talhões nos anos 10 e 20 para poder atender as restrições de adjacência para talhões vizinhos e distantes.

O problema com 50 talhões e 30 períodos também foi resolvido sem o uso de restrições de adjacência. Neste caso, obteve-se \$1.647,3 para o valor da função objetivo em 3 segundos.

Percebe-se neste caso que o uso das restrições de adjacência não acarretaram perda no valor da função objetivo, uma vez que para o uso das restrições de adjacência no mesmo período obtivemos \$1.647,1 para o valor da função objetivo, e na segunda situação observada, que é o uso de restrições de adjacência em períodos consecutivos tivemos \$1.646 para a função objetivo. A maior redução pode ser observada quando utilizamos restrições de adjacência para talhões vizinhos e distantes, neste caso a função objetivo caiu para \$1.618,9.

Esta redução pode ser compensada pela proximidade dos talhões que devem ser colhidos no mesmo período, pois esta proximidade deverá reduzir os custos de transporte, manutenção de estradas e deslocamento de pessoal.

## **CAPÍTULO VI**

### **6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS**

#### **6.1 CONCLUSÕES**

Dos vários testes executados, considerando diversos números de talhões e períodos no horizonte de planejamento, observou-se a validade da formulação proposta para as restrições de adjacência.

Foram utilizadas restrições de adjacência em três situações:

- 1ª) quando não se deseja colher talhões vizinhos em um mesmo período;
- 2ª) quando se pretende impedir a colheita de talhões vizinhos em períodos consecutivos;
- 3ª) quando se deseja impedir a colheita de talhões vizinhos e talhões distantes no mesmo período.

Nas três situações avaliadas a formulação desenvolvida mostrou-se eficaz, atendendo as exigências necessárias. Apesar do uso de restrições de adjacência aumentar o número de restrições do modelo de forma considerável, o tempo necessário para sua resolução não foi um fator dificultante nos testes efetuados. Uma empresa florestal de grande porte possui um número muito grande de talhões, porém suas fazendas estão distribuídas em vários municípios e geralmente tem uma equipe trabalhando em cada uma delas, neste caso para cada fazenda deve-se formular as restrições de adjacência. Caso a empresa não conte com uma equipe em cada fazenda faz-se uso das restrições de adjacência que impeçam a colheita de talhões vizinhos e talhões distantes em um mesmo

período, evitando assim que os funcionários precisem deslocar-se de uma fazenda para outra em um mesmo ano.

A utilização da matriz A para registro dos talhões vizinhos facilitou a formulação das restrições, pois, de acordo com as necessidades da empresa é que se definirá quais talhões são vizinhos e também quais são distantes.

A utilização do LINGO para a resolução do problema mostrou-se muito eficiente e de fácil manuseio e entendimento. O tempo computacional para a resolução dos problemas pelo LINGO pode ser considerado irrelevante levando-se em consideração as dimensões do problema. O LINGO é uma excelente ferramenta para resolução de problemas de forma rápida.

## **6.2 SUGESTÕES PARA TRABALHOS FUTUROS**

Uma possibilidade de continuação deste trabalho seria utilizar restrições de adjacência juntamente com um modelo de prognose de produção.

Também se poderia pensar no desenvolvimento de um programa computacional específico para este tipo de problema, dispensando assim a utilização do LINGO, que embora muito eficiente tem custo elevado.

No caso da utilização de restrições de adjacência para impedir a colheita de talhões vizinhos e distantes em um mesmo período pode-se procurar outra forma de registro dos talhões distantes que não seja somente 0 ou 1. Assim poderia-se considerar um peso para cada talhão distante, variando por exemplo entre 0 e 1, de acordo com a dificuldade de acesso entre os talhões ou mesmo a distância entre eles.

Outra alternativa seria trabalhar com a função objetivo considerando além do ganho financeiro com a maximização das receitas, essa função considerar também o ganho ambiental pelo fato de não estar-se devastando grandes áreas na mesma região.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. ANDRADE, E. L. **Introdução à pesquisa operacional – Métodos e modelos para a análise de decisão**. Rio de Janeiro, 2ª edição, 2002. Editora Livros Técnicos e Científicos.
2. BARROS, O. e WEINTRAUB, A. **Planning for a vertically integrated forest industry**. *Operations Research*, 30(6), 1982.
3. BLAZEWICZ J. et al. **Scheduling Computer and Manufacturing Process**. Springer-Verlag, Berlin, 1996.
4. BORGES, J. F. et al. **O planejamento da exploração em função das necessidades da fábrica**. In VI Seminário de atualização sobre sistemas de exploração e transporte florestal. FUPET, Curitiba – PR. Anais: 153 – 172.
5. BORGES, J. G. et al. **Combining a decomposition strategy with dynamic programming to solve spatially constrained Forest management scheduling problems**. *Forest Science*, 1999. 45(2): 201-212.
6. BOSTON, J. G. e BETTINGER, P. **An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems**. *Forest Science*, 1999. 45(2): 292-301.
7. BRAZ, E.M. **Um modelo de programação Linear para garantia do rendimento sustentado em pequena propriedade na floresta tropical**. Dissertação de mestrado, Curitiba, 2001. Setor de Ciências Agrárias – UFPR.
8. BUONGIORNO, J. e GILLESS, J. K. **Forest management and economics: a primer in quantitative methods**. New York, Collin Macmillan, 1987.
9. CARNIERI, C. **Planejamento florestal otimizado via redes de manejo**. Tese de doutorado, Campinas, 1989. UNICAMP.

10. CARNIERI, et al. **Um sistema de planejamento florestal.** In II Encontro de Planejamento florestal. Anais. Curitiba, 1991.
11. CINTRA, G. **First Fit Decreasing with Backtracking: Um algoritmo exato para o problema de corte unidimensional baseado na conjectura MIRUP.** In: XXXIII Simpósio Brasileiro de Pesquisa Operacional – A Pesquisa Operacional e o Meio Ambiente. Anais, Campos do Jordão, 2001. p.1390.
12. DYKSTRA, D. P. **Mathematical programming for natural resource management.** New York, McGraw – Hill, 1984.
13. HOGANSON, H. M. e BORGES, J. G. **Using Dynamic Programming and Overlapping Subproblems to address Adjacency in Large Harvest Scheduling problems.** Forest Science, 1998. 44(4): 526-538.
14. HOSOKAWA, R. T. **O uso de funções de forma de tronco em estudo de volumetria de espécies florestais.** In: Seminário sobre atualidades e perspectivas florestais. Curitiba, 1982, EMBRAPA/CNPQ. Anais p. 7-14.
15. JOHNSON, K. N. e SCHEURMAN, H. L. **Techniques for prescribing optimal timber harvest and investment under different objectives – discussion and synthesis.** Forest Science, monograph 18, 1977.
16. JONES, J. G. et al. **Formulating adjacency constraints in Linear Optimization Models for Scheduling Projects in Tactical Planning.** Forest Science, nov – 1991, p 1283 -1297.
17. LOESCH, C. e HEIN, N. **Pesquisa Operacional – fundamentos e modelos.** Blumenau, 1999. Editora FURB.
18. **Manual do LINGO.** Lindo Systems, Inc. N. Dayton, Chicago, Illinois. Outubro, 1999.

19. **MAYERLE, S. F. Programação Linear Inteira e Mista, Programação Linear Binária.** Disponível em: <http://www.eps.ufsc.br/>. Acesso em dezembro de 2002.
20. **MCDILL, M. E. e BRAZE, J. Using the branch and bound algorithm to solve Forest Planning Problems with Adjacency constraints.** *Forest Science*, 47 (3). 2001, p 403 – 418.
21. **MCDILL, M. E. e BRAZE, J. Comparing Adjacency formulations for randomly generated Forest planning with four age-class distributions.** *Forest Science*, 46 (3). 2000, p 423 – 436.
22. **OLIVEIRA, E. B. Um sistema computadorizado de prognose do crescimento e produção de *pinus taeda L.*, com critérios quantitativos para a avaliação técnica e econômica de regimes de manejo.** Tese de doutorado, Curitiba, 1995. Setor de Ciências Agrárias – UFPR.
23. **OLIVEIRA, E. B., MACHADO, S.A. e FIGUEIREDO FILHO, A. Sistema para simulação do crescimento e da produção de *Pinus taeda L.* e a avaliação econômica de regimes de manejo.** Curitiba: EMBRAPA-CNPQ, 1998.
24. **OLIVEIRA, E. B., MACHADO, S.A. e HOEFLICH, V.A. A análise econômica de regimes de manejo de pinus.** Curitiba: EMBRAPA-CNPQ, 1998., 51p. (Série documentos).
25. **PUCCHINI, A. L. e PIZZOLATO, N. Programação Linear.** Rio de Janeiro, 2ª edição, 1990. Editora Livros Técnicos e científicos.
26. **RENDER, B. e STAIR, R. M. Quantitative Analysis for Management.** Prentice-Hall Inc. New Jersey, 6ª ed., 1997.
27. **RIBEIRO FILHO, G. e LORENA, L. A. N. Algoritmo Genético Construtivo e Geração de Colunas: uma aplicação para coloração de grafos.** In. XXXII Simpósio Brasileiro de Pesquisa Operacional. Anais. Viçosa, 2000. p.1570-1580.

28. SANQUETTA, C. R. **Fundamentos biométricos dos modelos de simulação florestal.** FUDEF, Curitiba, 1996.
29. SCOLFORO, J. R. S. **Sistema integrado para predição e análise presente e futura do crescimento e produção, com otimização de remuneração de capitais, para Pinus Caribaea var. hondurensis.** Tese de doutorado, Curitiba, 1990. Setor de ciências agrárias – UFPR.
30. SCOLFORO, J. R. S. **Inventário florestal.** ESAL/FAEPE, Minas Gerais, 1993.
31. SPIECKER, H. **Métodos de pesquisa operacional com aplicações em manejo florestal.** Universidade Federal do Paraná, setor de Ciências Agrárias, curso de Engenharia florestal, 1975.
32. SOUZA, A. e CLEMENTE, A. **Decisões financeiras e análise de investimentos – Fundamentos, Técnicas e aplicações.** São Paulo, Atlas, 3ª edição, 1999.
33. VOLPI, N. M. P. **O impacto de perturbações estocásticas em um modelo de planejamento florestal.** Tese de doutorado, Curitiba, 1997. Setor de Ciências agrárias – UFPR.
34. WAGER, H. M. **Pesquisa Operacional.** Rio de Janeiro, 2ª ed. 1985. Prentice-Hall do Brasil Ltda.
35. WARE, G.O. e CLUTTER, J.L. **A mathematical programming system for the management of industrial forest.** Forest Science. Peking 17(4) 425 – 445. dec. 1971.
36. WEINTRAUB, A. et al. **A Column Generation Algorithm for Solving General Forest Planning Problems with Adjacency constraints.** Forest Science, fev. 1994; p 142 -161
37. ZIONTS, S. **Linear and integer programming.** Prentice-Hall Inc. New Jersey.

## ANEXO 1

### UTILIZAÇÃO DO LINGO

O LINGO (Language for Interactive General Optimizer) é um software desenvolvido por LINDO System's Inc. de manuseio amigável utilizado para resolver problemas de otimização envolvendo programação linear, programação linear inteira, programação linear binária, programação não-linear.

O LINGO permite formular problemas dos mais complexos, resolvê-los e analisar sua solução, reduzindo drasticamente o tempo de desenvolvimento e solução dos modelos de otimização. O LINGO possui um ambiente de modelagem interativa, linguagem de modelagem completa e opções flexíveis para o tratamento dos dados tornando-o assim, uma poderosa ferramenta para modelar problemas de qualquer complexidade.

O LINGO reconhece variáveis subscriptas, conjuntos, operações sobre conjuntos e expressões matemáticas genéricas. Com isso, os modelos poderão ser expressos de maneira concisa e de fácil leitura. O LINGO vem com uma biblioteca de funções matemáticas e estatísticas e permite que os dados sejam armazenados em arquivos externos e planilhas.

Para escrever um modelo no LINGO podemos utilizar sua área de trabalho e escrever o modelo explicitamente, ou também, utilizar o modelo de linguagem do LINGO, que não difere muito da linguagem matemática. Cada linha no LINGO deve ser terminada com ponto e vírgula (;), podendo-se quebrar uma expressão em várias linhas para sua melhor compreensão dentro do programa, porém para que a expressão seja considerada completa ela deve ser terminada com o ponto e vírgula.

Também pode-se inserir comentários no meio do modelo para alguma explicação necessária, esses comentários devem ser iniciados com ponto de exclamação (!) e terminados com ponto-e-vírgula (;), assim, todo texto entre ! e ; não será considerado pelo LINGO.

O LINGO não distingue letras maiúsculas de letras minúsculas, desta forma, a variável MESA = mesa. As variáveis devem ser iniciadas com uma letra (A - Z) e os caracteres subseqüentes podem ser os números (0 -9) ou sublinhado ( \_ ). Os nomes não devem ter mais de 32 caracteres na sua composição.

O LINGO utiliza-se dos caracteres >= para denotar  $\geq$  (maior ou igual) e <= para denotar  $\leq$  (menor ou igual) e \* (asterisco) para indicar a multiplicação.

A utilização de variáveis indexadas (matrizes e vetores) é fundamental em problemas de grande porte. Estas variáveis e seus índices devem ser definidos no LINGO na área para definição dos conjuntos (sets). Os valores dos coeficientes constantes também devem ser registrados na área para definição dos dados (data). Após a definição das variáveis e dos dados, passa-se para a formulação do problema. Para definir um modelo no LINGO deve-se seguir o seguinte esquema:

**MODEL:**

SETS:

(definição dos conjuntos)

ENDSETS

DATA:

(definição dos dados)

ENDDATA

(formulação do problema)

**END**

A utilização dos dois pontos após MODEL, SETS e DATA é imprescindível na formulação do problema e, após ENDSETS, ENDDATA e END não se utiliza nenhum sinal.

O LINGO trabalha com as funções de looping a seguir:

@FOR que é usada para restrições gerais para todos os membros de um conjunto.

@SUM que computa a soma de uma expressão sobre todos os membros do conjunto.

A definição MIN indica que a função objetivo deve ser minimizada e MAX é utilizada para a maximização.

Com relação as variáveis, utiliza-se a definição @GIN, para restringir uma variável como inteira não negativa, @BIN para variável binária, @FREE significa que a variável pode assumir qualquer valor real e @BND limita a variável entre valores fixados.

Para exemplificar a sintaxe do LINGO, utilizaremos o problema a seguir:

Uma empresa de produtos manufaturados fabrica quatro diferentes tipos de artigos metálicos, cada um dos quais deve ser usinado, polido e montado. As necessidades específicas de tempo de trabalho (em horas) de cada um dos produtos são as seguintes:

	Usinagem	Polimento	Montagem
Produto I	3	1	2
Produto II	2	1	1
Produto III	2	2	2
Produto IV	2,5	3	1

A empresa dispõe para si, numa base semanal, de 650 horas de tempo de usinagem, 450 horas de tempo de polimento e 500 horas de tempo de montagem. Os lucros unitários sobre os produtos são, respectivamente, \$6, \$5, \$6 e \$8. A empresa firmou um contrato com um distribuidor para fornecer-lhe, semanalmente, um mínimo de 50 unidades do produto I e 100 unidades de qualquer combinação dos produtos II e III. Por intermédio de outros clientes a empresa pode vender semanalmente tantas unidades quantas produza dos produtos I, II e III, mas apenas um máximo de 75 unidades do produto IV. Quantas unidades de cada produto a empresa deve manufaturar semanalmente a fim de atender às obrigações contratuais e maximizar seu lucro total? Admita ainda que a empresa não deseja formar estoques desnecessários dentro da semana.

#### **Modelagem do problema:**

Consideremos as variáveis  $x_i$  = unidades do produto  $i$  produzidas semanalmente, para  $i = 1, 2, 3, 4$ .

#### **Função objetivo (maximizar o lucro sobre a produção total)**

$$\text{Max } Z = 6x_1 + 5x_2 + 6x_3 + 8x_4$$

Sujeito a :

$$3x_1 + 2x_2 + 2x_3 + 2,5x_4 \leq 650 \quad (\text{tempo limite para usinagem})$$

$$x_1 + x_2 + 2x_3 + 3x_4 \leq 450 \quad (\text{tempo limite para polimento})$$

$$\begin{aligned}
 2x_1 + x_2 + 2x_3 + x_4 &\leq 500 && \text{(tempo limite para montagem)} \\
 x_1 &\geq 50 && \text{(contrato firmado para o produto I)} \\
 x_2 + x_3 &\geq 100 && \text{(contrato firmado para os produtos II e III)} \\
 x_4 &\leq 75 && \text{(venda máxima do produto IV)} \\
 x_1, x_2, x_3, x_4 &\geq 0 \text{ e inteiros.}
 \end{aligned}$$

Para a digitação da função objetivo na janela de trabalho do LINGO podemos escrever cada parcela como aparece na função objetivo, ou para evitar erros na digitação no caso de termos muitos valores, podemos utilizar a linguagem do LINGO. Matematicamente podemos escrever a função objetivo como  $Max \sum_{i=1}^4 l_i \cdot x_i$ , onde  $l_i$  é o lucro proporcionado pela venda de uma unidade do produto  $i$ . Assim:

! Função Objetivo;

```
max = @sum(produto(i) : l(i) * x(i) );
```

Observa-se que é necessário criar uma ligação entre o lucro de cada produto com sua produção, esta ligação foi designada por produto(i).

Depois de escrita a função objetivo passamos para as restrições, as três primeiras restrições do modelo referem-se ao tempo máximo disponível para cada processo, usinagem, polimento e montagem e podem ser escritas da seguinte forma:

```
@sum(produto(i) : u(i) * x(i) ) <= 650 ;
@sum(produto(i) : p(i) * x(i) ) <= 450 ;
@sum(produto(i) : m(i) * x(i) ) <= 500 ;
```

onde:

$u(i)$  é o tempo de usinagem na produção de uma unidade do produto  $i$

$p(i)$  é o tempo de polimento na produção de uma unidade do produto  $i$

$m(i)$  é o tempo de montagem na produção de uma unidade do produto  $i$

As três últimas restrições referem-se a demanda para cada tipo de produto e podem ser escritas da seguinte forma:

```
@for (produto (i) | i#eq#1 : x(i) >= 50) ;
@sum (produto (i) : d(i) * x(i) ) >= 100 ;
@for (produto (i) | i#eq#4 : x(i) <= 75) ;
```

Nestas restrições foi utilizado o operador lógico do LINGO, # eq # o qual indica que i deve ser igual a 1, na primeira restrição e igual a 4 na terceira, além desse operador lógico ainda temos:

**#NOT#** que é verdadeiro se o operador imediatamente da direita for verdadeiro, caso contrário é falso.

**#NE#** diferente

**#GT#** maior que

**#GE#** maior ou igual a

**#LT#** menor que

**#LE#** menor ou igual a

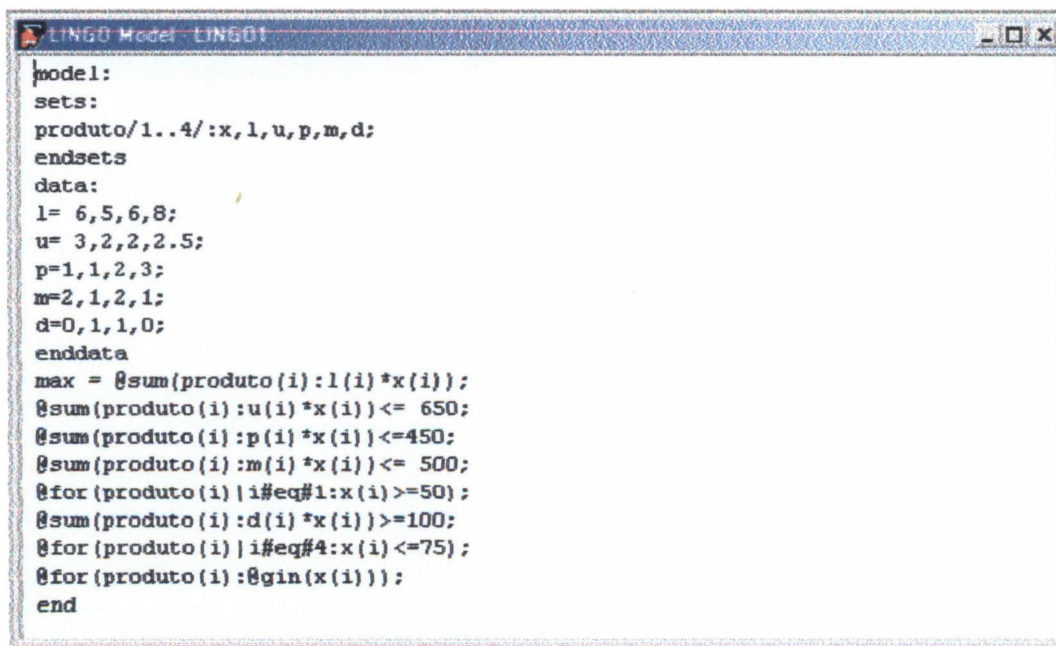
**#AND#** operador (e) que é verdadeiro se ambos os argumentos são verdadeiros e falso em qualquer outro caso.

**#OR#** operador (ou) que é falso somente se ambos os argumentos são falsos, em qualquer outro caso é verdadeiro.

Após a formulação das restrições precisamos definir as variáveis como inteiras, da seguinte forma:

```
@for (produto (i) : @gin (x(i) ) ) ;
```

A figura a seguir mostra a janela do LINGO com a toda a formulação do problema, desde a definição dos conjuntos, dados e a formulação propriamente dita:




```

LINGO Model - LINGO1
model:
sets:
produto/1..4/:x,l,u,p,m,d;
endsets
data:
l= 6,5,6,8;
u= 3,2,2,2.5;
p=1,1,2,3;
m=2,1,2,1;
d=0,1,1,0;
enddata
max = @sum(produto(i):l(i)*x(i));
@sum(produto(i):u(i)*x(i))<= 650;
@sum(produto(i):p(i)*x(i))<=450;
@sum(produto(i):m(i)*x(i))<= 500;
@for(produto(i)|i#eq#1:x(i)>=50);
@sum(produto(i):d(i)*x(i))>=100;
@for(produto(i)|i#eq#4:x(i)<=75);
@for(produto(i):@gin(x(i)));
end

```

Figura 1. Definição dos conjuntos, dados e a formulação de um problema no LINGO.

Após terminada a formulação do problema, o mesmo está pronto para ser resolvido, o que pode ser feito selecionando-se o comando Solve no LINGO menu, ou clicando-se o ícone  na barra de ferramentas. Com este procedimento o LINGO irá compilar o modelo e verificar se não existe nenhum erro de sintaxe, caso exista, ele abrirá uma janela identificando o tipo de erro ocorrido e onde ocorreu. Se não existir nenhum erro o LINGO começará a processar, para encontrar a solução ótima do modelo. Quando a resolução inicia-se é aberta uma janela Solver Status, para que se possa monitorar o processo de resolução e as dimensões do modelo. Esta janela mostrará o número total de variáveis, número de variáveis não lineares, número de variáveis inteiras. Em geral, quanto mais variáveis não-lineares e variáveis inteiras tiver o modelo mais dificuldade ele terá para ser resolvido, levando em consideração o tempo necessário para tal. Nesta mesma janela aparecem o número total de restrições e o número de restrições não lineares, também o número de coeficientes não nulos envolvidos no problema e o número desses que aparecem com variáveis não-lineares, e ainda o tempo utilizado para resolver o modelo e a memória requisitada para tal.

Nesta mesma janela temos o Optimizer Status Box que mostra o estado atual da otimização, neste campo aparecem as seguintes descrições:

**State:**

**Global Optimun** – quando o LINGO encontra uma solução ótima global para o problema.

**Local Optimun** – quando o LINGO encontra uma solução ótima local para o problema.

**Feasible** – o modelo é factível, porém o ótimo ainda não foi encontrado.

**Unbounded** – solução infinita.

**Interrupted** – a resolução foi interrompida

**Undertermined** – o estado do modelo não foi determinado até o momento.

**Nº de iterações** – apresenta o número de iterações realizadas até o momento.

**Infeasibility** – número de restrições que são violadas.

**Objective** – valor corrente da função objetivo

**Best IP** – valor da função objetivo para a melhor solução inteira encontrada.

**IP Bound** – limite teórico da função objetivo para modelos de programação inteira.

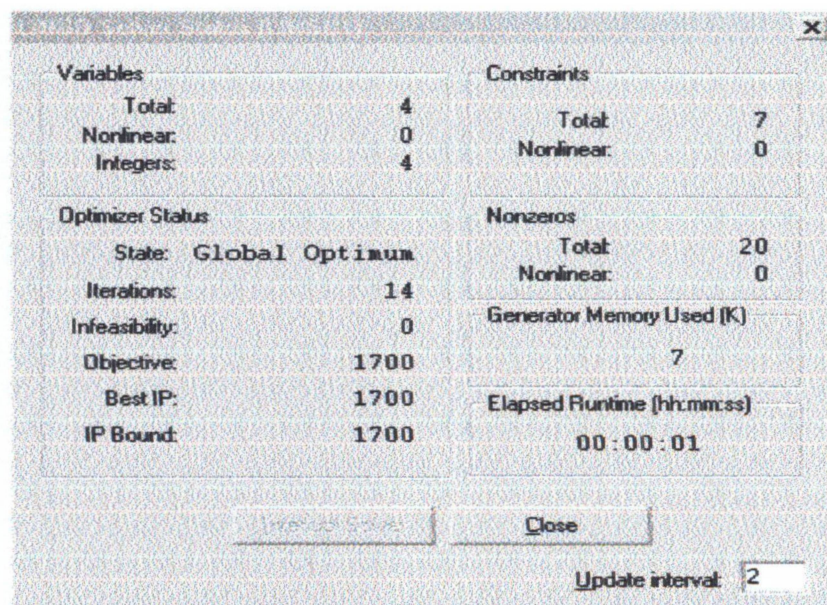


Figura 2. Janela Solver Status do LINGO

Terminado o processamento do problema e este tendo solução ótima clica-se em close para que seja apresentada a solução do problema. Caso deseje-se visualizar somente as variáveis não nulas da solução devemos clicar na barra de ferramentas **Window** e em seguida **Command Window** e digitar **nonz x** ↵ então será mostrada a seguinte resposta:

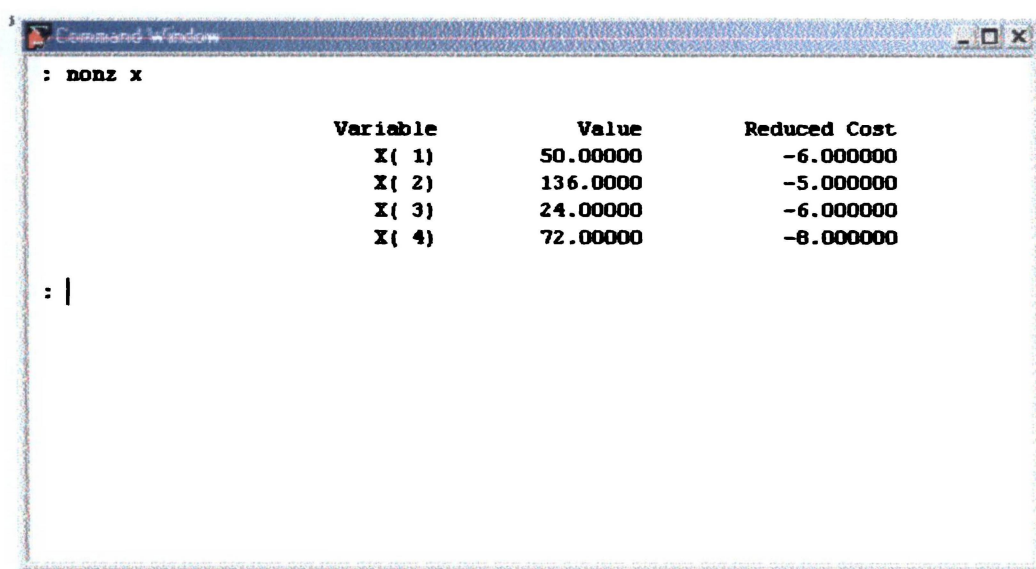


Figura 3. Solução do problema

A solução final do problema proposto é a seguinte: produzir 50 unidades do produto I, 136 unidades do produto II, 24 unidades do produto III e 72 unidades do produto IV com lucro total de \$1700. Observando a figura 5.2. verificamos que foi necessário 1 segundo para a solução do problema, e o mesmo continha 4 variáveis e 7 restrições lineares sendo resolvido com 14 iterações.

## ANEXO 2

**DADOS UTILIZADOS PARA O PROBLEMA COM 16 TALHÕES E 10 PERÍODOS**

As tabelas abaixo registram a receita anual propiciada por cada talhão e os volumes anuais de madeira para laminação (produto 1), serraria (produto 2) e processo (produto 3) para cada talhão.

	RECEITA ANUAL POR TALHÃO (\$1000)									
	1	2	3	4	5	6	7	8	9	10
1	250	300	395	465	555	635	735	890	1005	1125
2	100	120	158	186	222	245	294	346	392	440
3	100	120	158	186	222	245	294	346	392	440
4	200	240	310	372	444	508	588	712	804	900
5	200	240	310	372	444	508	588	712	804	900
6	75	90	118,5	139,5	166,5	190,5	220,5	267	301,5	450
7	500	600	790	930	1110	1270	1470	1780	2010	2250
8	375	450	592,5	697,5	832,5	952,5	1102,	1335	1507,	1687,5
9	500	600	790	930	1110	1270	1470	1780	2010	2250
10	375	450	592,5	697,5	832,5	952,5	1102,	1335	1507,	1687,5
11	300	360	474	558	660	762	882	1068	1206	1350
12	225	270	355,5	418,5	499,5	571,5	661,5	801	904,5	1012,5
13	300	360	474	558	660	762	882	1068	1206	1350
14	225	270	355,5	418,5	499,5	571,5	661,5	801	904,5	1012,5
15	225	270	355,5	418,5	499,5	571,5	661,5	801	904,5	1012,5
16	225	270	355,5	418,5	499,5	571,5	661,5	801	904,5	1012,5

Tabela 1: Receita propiciada por cada talhão em cada período.

<b>TALHÃO 1 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	5	50	75
2	10	55	85
3	20	65	100
4	25	75	115
5	30	85	150
6	35	95	175
7	45	105	195
8	55	130	225
9	65	145	245
10	75	160	270

Tabela 2: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 1.

<b>TALHÃO 2 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	2	20	30
2	4	22	34
3	8	26	40
4	10	30	46
5	12	34	60
6	14	38	70
7	18	42	78
8	20	52	90
9	24	58	98
10	28	64	108

Tabela 3: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 2.

<b>TALHÃO 3 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	2	20	30
2	4	22	34
3	8	26	40
4	10	30	46
5	12	34	60
6	14	38	70
7	18	42	78
8	20	52	90
9	24	58	98
10	28	64	108

Tabela 4: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 3.

<b>TALHÃO 4 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4	40	60
2	8	44	68
3	16	52	80
4	20	60	92
5	24	68	120
6	28	76	140
7	36	84	156
8	44	104	180
9	52	116	196
10	60	128	216

Tabela 5 Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 4.

<b>TALHÃO 5 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4	40	60
2	8	44	68
3	16	52	80
4	20	60	92
5	24	68	120
6	28	76	140
7	36	84	156
8	44	104	180
9	52	116	196
10	60	128	216

Tabela 6: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 5.

<b>TALHÃO 6 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	1,5	15	22,5
2	3	16,5	25,5
3	6	19,5	30
4	7,5	22,5	34,5
5	9	25,5	45
6	10,5	28,5	52,5
7	13,5	31,5	58,5
8	16,5	39	67,5
9	19,5	43,5	73,5
10	45	48	81

Tabela 7: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 6.

<b>TALHÃO 7 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	10	100	150
2	20	110	170
3	40	130	200
4	50	150	230
5	60	170	300
6	70	190	350
7	90	210	390
8	110	260	450
9	130	290	490
10	150	320	540

Tabela 8: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 7.

<b>TALHÃO 8 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	7,5	75	112,5
2	15	82,5	127,5
3	30	97,5	150
4	37,5	112,5	172,5
5	45	127,5	225
6	52,5	142,5	262,5
7	67,5	157,5	292,5
8	82,5	195	337,5
9	97,5	217,5	367,5
10	112,5	240	405

Tabela 9: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 8.

<b>TALHÃO 9 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	10	100	150
2	20	110	170
3	40	130	200
4	50	150	230
5	60	170	300
6	70	190	350
7	90	210	390
8	110	260	450
9	130	290	490
10	150	320	540

Tabela 10: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 9.

<b>TALHÃO 10 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	7,5	75	112,5
2	15	82,5	127,5
3	30	97,5	150
4	37,5	112,5	172,5
5	45	127,5	225
6	52,5	142,5	262,5
7	67,5	157,5	292,5
8	82,5	195	337,5
9	97,5	217,5	367,5
10	112,5	240	405

Tabela 11: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 10.

<b>TALHÃO 11 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	6	60	90
2	12	66	102
3	24	78	120
4	30	90	138
5	36	102	180
6	42	114	210
7	54	126	234
8	66	156	270
9	78	174	294
10	90	192	324

Tabela 12: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 11.

<b>TALHÃO 12 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4,5	45	67,5
2	9	49,5	76,5
3	18	58,5	90
4	22,5	67,5	103,3
5	27	76,5	135
6	31,5	85,5	157,5
7	40,5	94,5	175,5
8	49,5	117	202,5
9	58,5	130,5	220,5
10	67,5	144	243

Tabela 13: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 12.

<b>TALHÃO 13 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	6	60	90
2	12	66	102
3	24	78	120
4	30	90	138
5	36	102	180
6	42	114	210
7	54	126	234
8	66	156	270
9	78	174	294
10	90	192	324

Tabela 14: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 13.

<b>TALHÃO 14 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4,5	45	67,5
2	9	49,5	76,5
3	18	58,5	90
4	22,5	67,5	103,3
5	27	76,5	135
6	31,5	85,5	157,5
7	40,5	94,5	175,5
8	49,5	117	202,5
9	58,5	130,5	220,5
10	67,5	144	243

Tabela 15: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 14.

<b>TALHÃO 15 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4,5	45	67,5
2	9	49,5	76,5
3	18	58,5	90
4	22,5	67,5	103,3
5	27	76,5	135
6	31,5	85,5	157,5
7	40,5	94,5	175,5
8	49,5	117	202,5
9	58,5	130,5	220,5
10	67,5	144	243

Tabela 16: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 15.

<b>TALHÃO 16 (Volume em 1000 m<sup>3</sup>)</b>			
<b>Período</b>	<b>Produto 1</b>	<b>Produto 2</b>	<b>Produto 3</b>
1	4,5	45	67,5
2	9	49,5	76,5
3	18	58,5	90
4	22,5	67,5	103,3
5	27	76,5	135
6	31,5	85,5	157,5
7	40,5	94,5	175,5
8	49,5	117	202,5
9	58,5	130,5	220,5
10	67,5	144	243

Tabela 17: Quantidade em m<sup>3</sup> de madeira disponível por período e por produto no talhão 16

## ANEXO 3

**FORMULAÇÃO DOS PROBLEMAS DE PLANEJAMENTO FLORESTAL  
COM RESTRIÇÕES DE ADJACÊNCIA UTILIZANDO A SINTAXE DO  
LINGO**

**A - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos em um mesmo período**

**MODEL:**

**SETS:**

**TALHAO /1..16/;**

**TEMPO /1..10/;**

**PRODUTO/1..3/;**

**MATRIZ (TALHAO, TEMPO) :X, r;**

**VOLUME (TALHAO, TEMPO, PRODUTO) :V;**

**DEMANDA (TEMPO, PRODUTO) :D;**

**VIZINHOS (TALHAO, TALHAO) :A;**

**ENDSETS**

**DATA:**

**r =**

250,300,395,465,555,635,735,890,1005,1125,  
100,120,158,186,222,245,294,346,392,440,  
100,120,158,186,222,245,294,346,392,440,  
200,240,316,372,444,508,588,712,804,900,  
200,240,316,372,444,508,588,712,804,900,  
75,90,118.5,139.5,166.5,190.5,220.5,267,301.5,450,  
500,600,790,930,1110,1270,1470,1780,2010,2250,  
375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,  
500,600,790,930,1110,1270,1470,1780,2010,2250,  
375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,  
300,360,474,558,666,762,882,1068,1206,1350,  
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
300,360,474,558,666,762,882,1068,1206,1350,  
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5;

**V =**

5,50,75,10,55,85,20,65,100,25,75,115,30,85,150,35,95,175,45,105,19  
5,55,130,225,65,145,245,75,160,270,  
2,20,30,4,22,34,8,26,40,10,30,46,12,34,60,14,38,70,18,42,78,20,52,  
90,24,58,98,28,64,108,  
2,20,30,4,22,34,8,26,40,10,30,46,12,34,60,14,38,70,18,42,78,20,52,  
90,24,58,98,28,64,108,  
4,40,60,8,44,68,16,52,80,20,60,92,24,68,120,28,76,140,36,84,156,44  
,104,180,52,116,196,60,128,216,

4,40,60,8,44,68,16,52,80,20,60,92,24,68,120,28,76,140,36,84,156,44,  
 104,180,52,116,196,60,128,216,  
 1.5,15,22.5,3,16.5,25.5,6,19.5,30,7.5,22.5,34.5,9,25.5,45,10.5,28.  
 5,52.5,13.5,31.5,58.5,16.5,39,67.5,19.5,43.5,73.5,45,48,81,  
 10,100,150,20,110,170,40,130,200,50,150,230,60,170,300,70,190,350,  
 90,210,390,110,260,450,130,290,490,150,320,540,  
 7.5,75,112.5,15,82.5,127.5,30,97.5,150,37.5,112.5,172.5,45,127.5,2  
 25,52.5,142.5,262.5,67.5,157.5,292.5,82.5,195,337.5,97.5,217.5,367  
 .5,112.5,240,405,  
 10,100,150,20,110,170,40,130,200,50,150,230,60,170,300,70,190,350,  
 90,210,390,110,260,450,130,290,490,150,320,540,  
 7.5,75,112.5,15,82.5,127.5,30,97.5,150,37.5,112.5,172.5,45,127.5,2  
 25,52.5,142.5,262.5,67.5,157.5,292.5,82.5,195,337.5,97.5,217.5,367  
 .5,112.5,240,405,  
 6,60,90,12,66,102,24,78,120,30,90,138,36,102,180,42,114,210,54,126  
 ,234,66,156,270,78,174,294,90,192,324,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243,  
 6,60,90,12,66,102,24,78,120,30,90,138,36,102,180,42,114,210,54,126  
 ,234,66,156,270,78,174,294,90,192,324,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243;

A =

1,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,  
 1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,  
 0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,  
 0,0,1,1,0,1,1,0,0,0,0,0,0,0,0,0,  
 1,1,0,0,1,1,0,1,1,1,0,0,0,0,0,0,  
 0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,  
 0,0,0,1,0,1,1,0,0,1,1,0,0,0,0,0,  
 1,0,0,0,1,0,0,1,1,0,0,1,0,0,0,0,  
 0,0,0,0,1,0,0,1,1,1,0,0,1,0,0,0,  
 0,0,0,0,1,1,1,0,1,1,1,0,0,1,0,0,  
 0,0,0,0,0,0,1,0,0,1,1,0,0,0,1,1,  
 0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,0,  
 0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,0,  
 0,0,0,0,0,0,0,0,0,1,0,0,1,1,0,  
 0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,  
 0,0,0,0,0,0,0,0,0,1,0,0,1,1,1;

D =

5,40,60,15,40,60,15,40,60,20,60,90,25,60,90,30,60,90,30,60,90,50,1  
 00,120,65,100,150,95,150,200;

ENDDATA

! função objetivo:

MAX= @SUM (TALHAO(I) :@SUM(TEMPO(J) :r(I,J)\*X(I,J)));

```

! restrições de corte;

@FOR(TALHAO(I) : @SUM(TEMPO(J) : X(I, J) <= 1) ;

! restrições de demanda;

@FOR(TEMPO(J) : @FOR(PRODUTO(K) : @SUM(TALHAO(I) : V(I, J, K) * X(I, J) >= D(J, K)) ;

! restrições de adjacência;

@FOR(TEMPO(K) :
    @FOR(TALHAO(I) :
        @SUM(TALHAO(J) | J #NE# I :
X(J, K) * A(I, J) <= @SUM(TALHAO(J) : A(I, J) * (1 - X(I, K)) ;
        ) ;
    ) ;

! discriminação das variáveis como binárias;

@FOR(TEMPO(J) : @FOR(TALHAO(I) : @BIN(X(I, J)) ;

END

```

**B - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos em períodos consecutivos.**

**MODEL:**

```

SETS:
TALHAO /1..16/;
TEMPO /1..10/;
PRODUTO/1..3/;
MATRIZ (TALHAO, TEMPO) :X, r;
VOLUME (TALHAO, TEMPO, PRODUTO) :V;
DEMANDA (TEMPO, PRODUTO) :D;
VIZINHOS (TALHAO, TALHAO) :A;

```

**ENDSETS**

**DATA:**

```

r =
250,300,395,465,555,635,735,890,1005,1125,
100,120,158,186,222,245,294,346,392,440,
100,120,158,186,222,245,294,346,392,440,
200,240,316,372,444,508,588,712,804,900,
200,240,316,372,444,508,588,712,804,900,
75,90,118.5,139.5,166.5,190.5,220.5,267,301.5,450,
500,600,790,930,1110,1270,1470,1780,2010,2250,
375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,
500,600,790,930,1110,1270,1470,1780,2010,2250,
375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,
300,360,474,558,666,762,882,1068,1206,1350,
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,
300,360,474,558,666,762,882,1068,1206,1350,
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,
225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5;

```

V =

5, 50, 75, 10, 55, 85, 20, 65, 100, 25, 75, 115, 30, 85, 150, 35, 95, 175, 45, 105, 19  
 5, 55, 130, 225, 65, 145, 245, 75, 160, 270,  
 2, 20, 30, 4, 22, 34, 8, 26, 40, 10, 30, 46, 12, 34, 60, 14, 38, 70, 18, 42, 78, 20, 52,  
 90, 24, 58, 98, 28, 64, 108,  
 2, 20, 30, 4, 22, 34, 8, 26, 40, 10, 30, 46, 12, 34, 60, 14, 38, 70, 18, 42, 78, 20, 52,  
 90, 24, 58, 98, 28, 64, 108,  
 4, 40, 60, 8, 44, 68, 16, 52, 80, 20, 60, 92, 24, 68, 120, 28, 76, 140, 36, 84, 156, 44  
 , 104, 180, 52, 116, 196, 60, 128, 216,  
 4, 40, 60, 8, 44, 68, 16, 52, 80, 20, 60, 92, 24, 68, 120, 28, 76, 140, 36, 84, 156, 44  
 , 104, 180, 52, 116, 196, 60, 128, 216,  
 1.5, 15, 22.5, 3, 16.5, 25.5, 6, 19.5, 30, 7.5, 22.5, 34.5, 9, 25.5, 45, 10.5, 28.  
 5, 52.5, 13.5, 31.5, 58.5, 16.5, 39, 67.5, 19.5, 43.5, 73.5, 45, 48, 81,  
 10, 100, 150, 20, 110, 170, 40, 130, 200, 50, 150, 230, 60, 170, 300, 70, 190, 350,  
 90, 210, 390, 110, 260, 450, 130, 290, 490, 150, 320, 540,  
 7.5, 75, 112.5, 15, 82.5, 127.5, 30, 97.5, 150, 37.5, 112.5, 172.5, 45, 127.5, 2  
 25, 52.5, 142.5, 262.5, 67.5, 157.5, 292.5, 82.5, 195, 337.5, 97.5, 217.5, 367  
 .5, 112.5, 240, 405,  
 10, 100, 150, 20, 110, 170, 40, 130, 200, 50, 150, 230, 60, 170, 300, 70, 190, 350,  
 90, 210, 390, 110, 260, 450, 130, 290, 490, 150, 320, 540,  
 7.5, 75, 112.5, 15, 82.5, 127.5, 30, 97.5, 150, 37.5, 112.5, 172.5, 45, 127.5, 2  
 25, 52.5, 142.5, 262.5, 67.5, 157.5, 292.5, 82.5, 195, 337.5, 97.5, 217.5, 367  
 .5, 112.5, 240, 405,  
 6, 60, 90, 12, 66, 102, 24, 78, 120, 30, 90, 138, 36, 102, 180, 42, 114, 210, 54, 126  
 , 234, 66, 156, 270, 78, 174, 294, 90, 192, 324,  
 4.5, 45, 67.5, 9, 49.5, 76.5, 18, 58.5, 90, 22.5, 67.5, 103.5, 27, 76.5, 135, 31.  
 5, 85.5, 157.5, 40.5, 94.5, 175.5, 49.5, 117, 202.5, 58.5, 130.5, 220.5, 67.5,  
 144, 243,  
 6, 60, 90, 12, 66, 102, 24, 78, 120, 30, 90, 138, 36, 102, 180, 42, 114, 210, 54, 126  
 , 234, 66, 156, 270, 78, 174, 294, 90, 192, 324,  
 4.5, 45, 67.5, 9, 49.5, 76.5, 18, 58.5, 90, 22.5, 67.5, 103.5, 27, 76.5, 135, 31.  
 5, 85.5, 157.5, 40.5, 94.5, 175.5, 49.5, 117, 202.5, 58.5, 130.5, 220.5, 67.5,  
 144, 243,  
 4.5, 45, 67.5, 9, 49.5, 76.5, 18, 58.5, 90, 22.5, 67.5, 103.5, 27, 76.5, 135, 31.  
 5, 85.5, 157.5, 40.5, 94.5, 175.5, 49.5, 117, 202.5, 58.5, 130.5, 220.5, 67.5,  
 144, 243;

A =

1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,  
 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0,  
 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0,  
 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,  
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1;

```
D =
5,40,60,15,40,60,15,40,60,20,60,90,25,60,90,30,60,90,30,60,90,50,1
00,120,65,100,150,95,150,200;
```

```
ENDDATA
```

```
!função objetivo;
```

```
MAX= @SUM (TALHAO(I):@SUM(TEMPO(J):r(I,J)*X(I,J)));
```

```
! restrição de corte;
```

```
@FOR (TALHAO(I):@SUM(TEMPO(J):X(I,J))<=1);
```

```
! restrições de demanda;
```

```
@FOR(TEMPO(J):@FOR(PRODUTO(K):@SUM(TALHAO(I):V(I,J,K)*X(I,J))>=D(J,K)));
```

```
! restrições de adjacência;
```

```
@for(tempo(k):
```

```
@for(talhao(i):
```

```
@sum(talhao(j)|j#ne# i:
```

```
x(j,k)*a(i,j))<=@sum(talhao(j):a(i,j)*(1-x(i,k)));
```

```
);
```

```
);
```

```
! restrições de adjacência;
```

```
@for(tempo(k)| k #ne# 10:
```

```
@for(talhao(i):
```

```
@sum(talhao(j)|j#ne#i:
```

```
x(j,k+1)*a(i,j))<=@sum(talhao(j):a(i,j)*(1-x(i,k)));
```

```
);
```

```
);
```

```
! discriminação das variáveis como binárias;
```

```
@FOR(TEMPO(J):@FOR(TALHAO(I):@BIN(X(I,J))));
```

```
END
```

**C - Restrições de adjacência aplicadas para impedir a colheita de talhões vizinhos e talhões distantes em um mesmo período.**

```
MODEL:
```

```
SETS:
```

```
TALHAO /1..16/;
```

```
TEMPO /1..10/;
```

```
PRODUTO/1..3/;
```

```
MATRIZ (TALHAO, TEMPO) :X, r;
```

```
VOLUME (TALHAO, TEMPO, PRODUTO) :V;
```

```
DEMANDA (TEMPO, PRODUTO) :D;
```

```
VIZINHOS (TALHAO, TALHAO) :A;
```

```
ENDSETS
```

DATA:

r =

250,300,395,465,555,635,735,890,1005,1125,  
 100,120,158,186,222,245,294,346,392,440,  
 100,120,158,186,222,245,294,346,392,440,  
 200,240,316,372,444,508,588,712,804,900,  
 200,240,316,372,444,508,588,712,804,900,  
 75,90,118.5,139.5,166.5,190.5,220.5,267,301.5,450,  
 500,600,790,930,1110,1270,1470,1780,2010,2250,  
 375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,  
 500,600,790,930,1110,1270,1470,1780,2010,2250,  
 375,450,592.5,697.5,832.5,952.5,1102.5,1335,1507.5,1687.5,  
 300,360,474,558,666,762,882,1068,1206,1350,  
 225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
 300,360,474,558,666,762,882,1068,1206,1350,  
 225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
 225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5,  
 225,270,355.5,418.5,499.5,571.5,661.5,801,904.5,1012.5;

v =

5,50,75,10,55,85,20,65,100,25,75,115,30,85,150,35,95,175,45,105,19  
 5,55,130,225,65,145,245,75,160,270,  
 2,20,30,4,22,34,8,26,40,10,30,46,12,34,60,14,38,70,18,42,78,20,52,  
 90,24,58,98,28,64,108,  
 2,20,30,4,22,34,8,26,40,10,30,46,12,34,60,14,38,70,18,42,78,20,52,  
 90,24,58,98,28,64,108,  
 4,40,60,8,44,68,16,52,80,20,60,92,24,68,120,28,76,140,36,84,156,44  
 ,104,180,52,116,196,60,128,216,  
 4,40,60,8,44,68,16,52,80,20,60,92,24,68,120,28,76,140,36,84,156,44  
 ,104,180,52,116,196,60,128,216,  
 1.5,15,22.5,3,16.5,25.5,6,19.5,30,7.5,22.5,34.5,9,25.5,45,10.5,28.  
 5,52.5,13.5,31.5,58.5,16.5,39,67.5,19.5,43.5,73.5,45,48,81,  
 10,100,150,20,110,170,40,130,200,50,150,230,60,170,300,70,190,350,  
 90,210,390,110,260,450,130,290,490,150,320,540,  
 7.5,75,112.5,15,82.5,127.5,30,97.5,150,37.5,112.5,172.5,45,127.5,2  
 25,52.5,142.5,262.5,67.5,157.5,292.5,82.5,195,337.5,97.5,217.5,367  
 .5,112.5,240,405,  
 10,100,150,20,110,170,40,130,200,50,150,230,60,170,300,70,190,350,  
 90,210,390,110,260,450,130,290,490,150,320,540,  
 7.5,75,112.5,15,82.5,127.5,30,97.5,150,37.5,112.5,172.5,45,127.5,2  
 25,52.5,142.5,262.5,67.5,157.5,292.5,82.5,195,337.5,97.5,217.5,367  
 .5,112.5,240,405,  
 6,60,90,12,66,102,24,78,120,30,90,138,36,102,180,42,114,210,54,126  
 ,234,66,156,270,78,174,294,90,192,324,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243,  
 6,60,90,12,66,102,24,78,120,30,90,138,36,102,180,42,114,210,54,126  
 ,234,66,156,270,78,174,294,90,192,324,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243,  
 4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.  
 5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,  
 144,243,

4.5,45,67.5,9,49.5,76.5,18,58.5,90,22.5,67.5,103.5,27,76.5,135,31.5,85.5,157.5,40.5,94.5,175.5,49.5,117,202.5,58.5,130.5,220.5,67.5,144,243;

A =

```
1,1,0,0,1,0,0,1,0,0,1,0,0,0,1,1,
1,1,1,0,1,1,0,0,0,0,0,0,0,0,0,0,
0,1,1,1,0,1,0,0,0,0,0,1,0,0,0,0,
0,0,1,1,0,1,1,0,0,0,0,1,0,0,0,0,
1,1,0,0,1,1,0,1,1,1,0,0,0,0,1,1,
0,1,1,1,1,1,0,0,1,0,0,0,0,0,0,0,
0,0,0,1,0,1,1,0,0,1,1,0,0,0,0,0,
1,0,0,0,1,0,0,1,1,0,0,1,0,0,0,0,
0,0,0,0,1,0,0,1,1,1,0,0,1,0,0,0,
0,0,0,0,1,1,1,0,1,1,1,0,0,1,0,0,
1,0,0,0,0,0,1,0,0,1,1,0,0,0,1,1,
0,0,1,1,0,0,0,1,0,0,0,1,1,0,0,0,
0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,0,
0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,0,
1,0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,
1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1;
```

D =

```
5,40,60,15,40,60,15,40,60,20,60,90,25,60,90,30,60,90,30,60,90,50,100,120,65,100,150,95,150,200;
```

ENDDATA

! função objetivo;

```
MAX= @SUM (TALHAO(I):@SUM(TEMPO(J):r(I,J)*X(I,J)));
```

! restrição de corte;

```
@FOR (TALHAO(I):@SUM(TEMPO(J):X(I,J))<=1);
```

! restrições de demanda;

```
@FOR (TEMPO(J):@FOR (PRODUTO(K):@SUM(TALHAO(I):v(I,J,K)*X(I,J))>=D(J,K)));
```

! restrições de adjacência;

```
@for (tempo(k):
```

```
  @for (talhao(i):
```

```
    @sum (talhao(j)| j #neq i:
```

```
      x(j,k)*a(i,j))<=@sum (talhao(j):a(i,j)*(1-x(i,k)));
```

```
    );
```

```
  );
```

! discriminação das variáveis como binárias;

```
@FOR (TEMPO(J):@FOR (TALHAO(I):@BIN(X(I,J))));
```

END

**ANEXO 4****DISTRIBUIÇÃO DOS TALHÕES UTILIZADOS NAS SIMULAÇÕES E SOLUÇÃO DE CADA SIMULAÇÃO**

Consideremos uma floresta composta por 50 talhões e horizonte de planejamento de 30 anos. A distribuição dos talhões seguem a figura abaixo.

1	11	21	31	41
2	12	22	32	42
3	13	23	33	43
4	14	24	34	44
5	15	25	35	45
6	16	26	36	46
7	17	27	37	47
8	18	28	38	48
9	19	29	39	49
10	20	30	40	50

Figura 4: Distribuição dos 50 talhões da floresta.

A seguir temos a solução encontrada para esta situação considerando restrições de adjacência para impedir a colheita de talhões vizinhos em um mesmo período (solução 1), em períodos consecutivos (solução 2) e também as utilizando para impedir que talhões vizinhos e talhões distantes sejam cortados em um mesmo período (solução 3).

**1) Solução 1****Função objetivo: 1647,1****Tempo: 6 segundos****Iterações: 548**

: nonz x

Variable	Value	Reduced Cost
X( 1, 30)	1.000000	-34.20000
X( 2, 9)	1.000000	-32.20000
X( 3, 29)	1.000000	-33.10000
X( 4, 30)	1.000000	-43.20000
X( 5, 23)	1.000000	-22.70000
X( 6, 1)	1.000000	-16.00000
X( 7, 30)	1.000000	-43.20000
X( 8, 15)	1.000000	-33.70000
X( 9, 2)	1.000000	-18.70000
X( 10, 20)	1.000000	-35.20000
X( 11, 20)	1.000000	-33.90000
X( 12, 28)	1.000000	-33.90000
X( 13, 26)	1.000000	-35.30000
X( 14, 12)	1.000000	-18.70000
X( 15, 30)	1.000000	-39.20000
X( 16, 25)	1.000000	-33.70000
X( 17, 11)	1.000000	-16.00000
X( 18, 30)	1.000000	-34.20000
X( 19, 10)	1.000000	-35.20000
X( 20, 30)	1.000000	-36.00000
X( 21, 30)	1.000000	-43.20000
X( 22, 13)	1.000000	-22.70000
X( 23, 7)	1.000000	-35.70000
X( 24, 30)	1.000000	-43.20000
X( 25, 17)	1.000000	-36.40000
X( 26, 8)	1.000000	-37.00000
X( 27, 19)	1.000000	-32.20000
X( 28, 10)	1.000000	-33.80000
X( 29, 18)	1.000000	-33.60000
X( 30, 24)	1.000000	-27.80000
X( 31, 3)	1.000000	-22.70000
X( 32, 30)	1.000000	-39.20000
X( 33, 27)	1.000000	-36.40000
X( 34, 5)	1.000000	-33.70000
X( 35, 30)	1.000000	-35.20000
X( 36, 20)	1.000000	-33.80000
X( 37, 30)	1.000000	-39.00000
X( 38, 20)	1.000000	-43.20000
X( 39, 14)	1.000000	-27.80000
X( 40, 30)	1.000000	-39.00000
X( 41, 30)	1.000000	-43.20000
X( 42, 4)	1.000000	-27.80000
X( 43, 30)	1.000000	-34.20000
X( 44, 10)	1.000000	-35.20000
X( 45, 22)	1.000000	-11.00000
X( 46, 30)	1.000000	-43.20000
X( 47, 16)	1.000000	-35.30000
X( 48, 6)	1.000000	-34.10000

X( 49, 30)	1.000000	-43.20000
X( 50, 21)	1.000000	-16.00000

## 2) Solução 2:

**Função Objetivo: 1646**

**Tempo: 14 segundos**

**Iterações: 1613**

: nonz x

Cost	Variable	Value	Reduced
	X( 1, 30)	1.000000	-34.20000
	X( 2, 10)	1.000000	-35.20000
	X( 3, 28)	1.000000	-29.60000
	X( 4, 30)	1.000000	-43.20000
	X( 5, 14)	1.000000	-27.80000
	X( 6, 1)	1.000000	-16.00000
	X( 7, 30)	1.000000	-43.20000
	X( 8, 24)	1.000000	-27.80000
	X( 9, 2)	1.000000	-18.70000
	X( 10, 20)	1.000000	-35.20000
	X( 11, 20)	1.000000	-33.90000
	X( 12, 30)	1.000000	-39.20000
	X( 13, 25)	1.000000	-33.70000
	X( 14, 12)	1.000000	-18.70000
	X( 15, 30)	1.000000	-39.20000
	X( 16, 23)	1.000000	-22.70000
	X( 17, 11)	1.000000	-16.00000
	X( 18, 30)	1.000000	-34.20000
	X( 19, 9)	1.000000	-32.20000
	X( 20, 30)	1.000000	-36.00000
	X( 21, 30)	1.000000	-43.20000
	X( 22, 16)	1.000000	-35.30000
	X( 23, 6)	1.000000	-34.10000
	X( 24, 30)	1.000000	-43.20000
	X( 25, 15)	1.000000	-33.70000
	X( 26, 8)	1.000000	-37.00000
	X( 27, 19)	1.000000	-32.20000
	X( 28, 10)	1.000000	-33.80000
	X( 29, 18)	1.000000	-33.60000
	X( 30, 26)	1.000000	-35.30000
	X( 31, 4)	1.000000	-27.80000
	X( 32, 30)	1.000000	-39.20000
	X( 33, 27)	1.000000	-36.40000
	X( 34, 5)	1.000000	-33.70000
	X( 35, 30)	1.000000	-35.20000
	X( 36, 20)	1.000000	-33.80000
	X( 37, 29)	1.000000	-36.10000
	X( 38, 20)	1.000000	-43.20000
	X( 39, 3)	1.000000	-22.70000

X( 40, 30)	1.000000	-39.00000
X( 41, 20)	1.000000	-43.20000
X( 42, 17)	1.000000	-36.40000
X( 43, 30)	1.000000	-34.20000
X( 44, 10)	1.000000	-35.20000
X( 45, 22)	1.000000	-11.00000
X( 46, 30)	1.000000	-43.20000
X( 47, 13)	1.000000	-22.70000
X( 48, 7)	1.000000	-35.70000
X( 49, 30)	1.000000	-43.20000
X( 50, 21)	1.000000	-16.00000

### 3) Solução 3:

**Função Objetivo: 1618,9**

**Tempo: 10 minutos e 45 segundos**

**Iterações: 41562**

: nonz x

Variable	Value	Reduced Cost
X( 1, 29)	1.000000	-29.20000
X( 2, 10)	1.000000	-35.20000
X( 3, 29)	1.000000	-33.10000
X( 4, 21)	1.000000	-12.10000
X( 5, 25)	1.000000	-33.70000
X( 6, 1)	1.000000	-16.00000
X( 7, 29)	1.000000	-39.20000
X( 8, 10)	1.000000	-43.20000
X( 9, 2)	1.000000	-18.70000
X( 10, 10)	1.000000	-34.20000
X( 11, 10)	1.000000	-33.80000
X( 12, 27)	1.000000	-31.00000
X( 13, 10)	1.000000	-43.20000
X( 14, 11)	1.000000	-16.00000
X( 15, 18)	1.000000	-33.60000
X( 16, 10)	1.000000	-43.20000
X( 17, 12)	1.000000	-18.70000
X( 18, 28)	1.000000	-25.90000
X( 19, 10)	1.000000	-35.20000
X( 20, 22)	1.000000	-11.00000
X( 21, 20)	1.000000	-39.20000
X( 22, 17)	1.000000	-36.40000
X( 23, 7)	1.000000	-35.70000
X( 24, 20)	1.000000	-39.20000
X( 25, 14)	1.000000	-27.80000
X( 26, 8)	1.000000	-37.00000
X( 27, 20)	1.000000	-35.20000
X( 28, 19)	1.000000	-30.70000
X( 29, 20)	1.000000	-39.00000
X( 30, 24)	1.000000	-27.80000
X( 31, 3)	1.000000	-22.70000

X( 32, 30)	1.000000	-39.20000
X( 33, 26)	1.000000	-35.30000
X( 34, 4)	1.000000	-27.80000
X( 35, 30)	1.000000	-35.20000
X( 36, 20)	1.000000	-33.80000
X( 37, 30)	1.000000	-39.00000
X( 38, 20)	1.000000	-43.20000
X( 39, 5)	1.000000	-33.70000
X( 40, 30)	1.000000	-39.00000
X( 41, 30)	1.000000	-43.20000
X( 42, 13)	1.000000	-22.70000
X( 43, 30)	1.000000	-34.20000
X( 44, 9)	1.000000	-32.20000
X( 45, 23)	1.000000	-14.00000
X( 46, 30)	1.000000	-43.20000
X( 47, 16)	1.000000	-35.30000
X( 48, 6)	1.000000	-34.10000
X( 49, 30)	1.000000	-43.20000
X( 50, 15)	1.000000	-33.70000

A seguir mostramos a distribuição dos 32 talhões utilizados nas simulações.

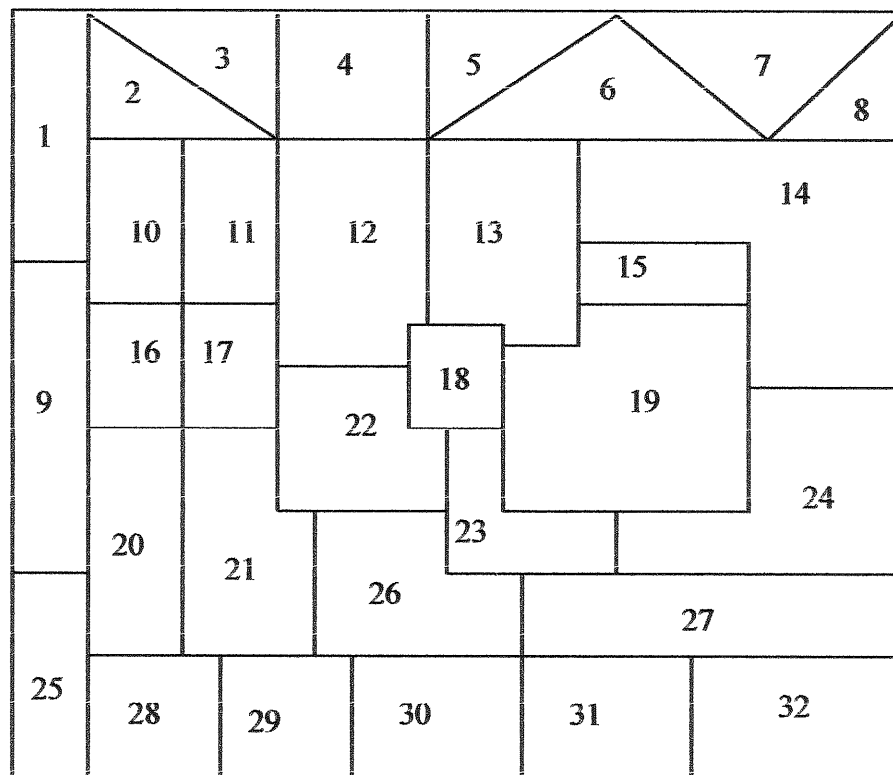


Figura 5: Descrição dos 32 talhões para simulação.

As soluções das três simulações para estes talhões estão apresentadas nas tabelas a seguir:

Período	Talhões selecionados para corte (restrições de adjacência para o mesmo período)
1	32
2	22
3	12
4	14
5	16 – 30
6	11
7	21 – 31
8	19
9	9 – 15 – 23
10	1 – 3 – 5 – 7 – 13 – 17 – 25 – 27 – 29
11	10
12	20
13	18 – 28
14	6 – 24
15	2 – 4 – 8 – 26

Tabela 18: Solução da simulação para 32 talhões e 15 anos, com restrições de adjacência no mesmo período.

Período	Talhões selecionados para corte (restrições de adjacência para períodos consecutivos)
1	22
2	32
3	12
4	14
5	16 – 30
6	15 – 21
7	1 – 11
8	19 – 27
9	9
10	3 – 5 – 7 – 13 – 17 – 25 – 29 – 31
11	10
12	20
13	6 – 18
14	26 – 28
15	2 – 4 – 8 – 24

Tabela 19: Solução da simulação para 32 talhões e 15 anos com restrições de adjacência para períodos consecutivos.

Período	Talhões selecionados para corte (restrições de adjacência para talhos vizinhos e distantes)
1	32
2	22
3	12
4	14
5	16 – 30
6	11
7	15 – 31
8	9 – 19 – 23 – 29
9	21 – 25 – 27
10	1 – 3 – 5 – 7 – 13 – 17
11	10
12	20
13	18 – 28
14	24 – 26
15	2 – 4 – 6 – 8

Tabela 20: Solução da simulações para 32 talhões e 15 anos com restrições de adjacência para talhões vizinhos e talhões distantes.

A próxima figura mostra a distribuição dos 25 talhões de uma floresta.

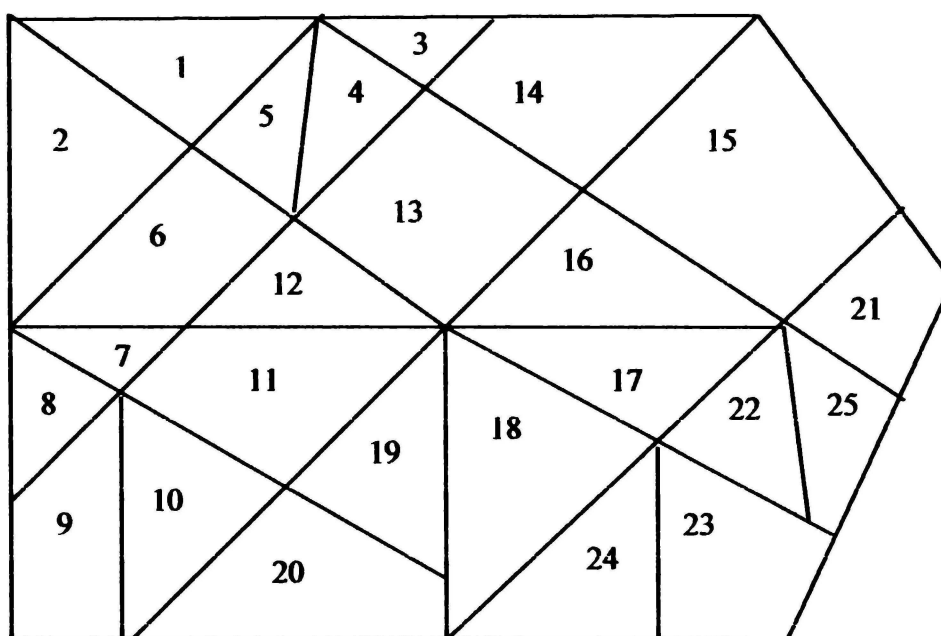


Figura 6: Distribuição dos 25 talhões da floresta.

A seguir temos a solução encontrada para a simulação com a floresta representada na figura 3, com 15 anos de horizonte de planejamento e restrições de adjacência impostas a talhões vizinhos em um mesmo período.

Variable	Value	Reduced Cost
X( 1, 8)	1.000000	-17.80000
X( 2, 13)	1.000000	-25.90000
X( 3, 10)	1.000000	-35.20000
X( 4, 12)	1.000000	-24.30000
X( 5, 10)	1.000000	-33.90000
X( 6, 4)	1.000000	-30.70000
X( 7, 10)	1.000000	-39.00000
X( 8, 15)	1.000000	-43.20000
X( 9, 10)	1.000000	-43.20000
X( 10, 1)	1.000000	-35.30000
X( 11, 7)	1.000000	-35.70000
X( 12, 2)	1.000000	-24.40000
X( 13, 9)	1.000000	-36.10000
X( 14, 3)	1.000000	-33.90000
X( 15, 10)	1.000000	-31.40000
X( 16, 14)	1.000000	-30.70000
X( 17, 15)	1.000000	-20.10000
X( 18, 12)	1.000000	-30.40000
X( 19, 14)	1.000000	-20.00000
X( 20, 10)	1.000000	-43.20000
X( 21, 13)	1.000000	-38.20000
X( 22, 6)	1.000000	-35.30000
X( 23, 15)	1.000000	-43.20000
X( 24, 5)	1.000000	-33.70000
X( 25, 11)	1.000000	-34.10000

Também apresentamos a solução para a mesma situação com restrições de adjacência impostas a talhões vizinhos em anos consecutivos.

Variable	Value	Reduced Cost
X( 1, 8)	1.000000	-17.80000
X( 2, 13)	1.000000	-25.90000
X( 3, 10)	1.000000	-35.20000
X( 4, 12)	1.000000	-24.30000
X( 5, 10)	1.000000	-33.90000
X( 6, 2)	1.000000	-24.40000
X( 7, 10)	1.000000	-39.00000
X( 8, 15)	1.000000	-43.20000
X( 9, 10)	1.000000	-43.20000
X( 10, 1)	1.000000	-35.30000
X( 11, 7)	1.000000	-35.70000
X( 12, 4)	1.000000	-30.70000
X( 13, 9)	1.000000	-36.10000
X( 14, 3)	1.000000	-33.90000

X( 15, 10)	1.000000	-31.40000
X( 16, 13)	1.000000	-26.60000
X( 17, 15)	1.000000	-20.10000
X( 18, 12)	1.000000	-30.40000
X( 19, 14)	1.000000	-20.00000
X( 20, 10)	1.000000	-43.20000
X( 21, 14)	1.000000	-40.20000
X( 22, 6)	1.000000	-35.30000
X( 23, 15)	1.000000	-43.20000
X( 24, 5)	1.000000	-33.70000
X( 25, 11)	1.000000	-34.10000

A última simulação efetuada para estes talhões foi para testar as restrições de adjacência impostas a talhões vizinhos e distantes. A solução encontrada foi a seguinte:

Cost	Variable	Value	Reduced
	X( 1, 8)	1.000000	-17.80000
	X( 2, 15)	1.000000	-34.20000
	X( 3, 9)	1.000000	-32.20000
	X( 4, 12)	1.000000	-24.30000
	X( 5, 15)	1.000000	-16.60000
	X( 6, 12)	1.000000	-26.40000
	X( 7, 10)	1.000000	-39.00000
	X( 8, 15)	1.000000	-43.20000
	X( 9, 10)	1.000000	-43.20000
	X( 10, 1)	1.000000	-35.30000
	X( 11, 6)	1.000000	-34.10000
	X( 12, 15)	1.000000	-36.00000
	X( 13, 10)	1.000000	-39.00000
	X( 14, 2)	1.000000	-31.00000
	X( 15, 3)	1.000000	-10.10000
	X( 16, 14)	1.000000	-30.70000
	X( 17, 10)	1.000000	-33.30000
	X( 18, 13)	1.000000	-33.60000
	X( 19, 7)	1.000000	-31.00000
	X( 20, 10)	1.000000	-43.20000
	X( 21, 13)	1.000000	-38.20000
	X( 22, 4)	1.000000	-27.80000
	X( 23, 14)	1.000000	-39.20000
	X( 24, 5)	1.000000	-33.70000
	X( 25, 11)	1.000000	-34.10000