

WESLEY PEREIRA MARCOS

**Cadeia de Markov aplicada ao manejo de pragas em
lavoura cafeeira**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA

2014

WESLEY PEREIRA MARCOS

**Cadeia de Markov aplicada ao manejo de pragas em lavoura
cafeeira**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Sólidos e Vibrações

Orientador: Prof. Dr. José Jean-Paul Zanlucchi de Souza Tavares

UBERLÂNDIA - MG

2014

WESLEY PEREIRA MARCOS

CADEIA DE MARKOV APLICADA AO MANEJO DE PRAGAS EM LAVOURA CAFEEIRA

Dissertação **aprovada** pelo Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia.

Área de concentração: Mecânica dos Sólidos e Vibrações

Banca Examinadora:

Prof. Dr. José Jean Paul Zanlucchi de Souza Tavares – UFU – Orientador

Prof. Dr. Marcos Antônio Viana Duarte - UFU

Prof. Dr. Marcius Fabius Henrique Carvalho - PUCCAMP

Uberlândia, 19 de Dezembro 2014

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

M321c Marcos, Wesley Pereira, 1989-
2014 Cadeia de Markov aplicada ao manejo de pragas em lavoura cafeeira
/ Wesley Pereira Marcos. - 2014.
145 f. : il.

Orientador: José Jean-Paul Zanlucchi de Souza Tavares.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Mecânica.
Inclui bibliografia.

1. Engenharia mecânica - Teses. 2. Markov, Processos de - Teses. 3.
Sistemas estocásticos - Teses. 4. Pragas - Controle - Teses. I. Tavares,
José Jean-Paul Zanlucchi de Souza, 1962- II. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. III.
Titulo.

CDU: 621

DEDICATÓRIA

“A todos que tornaram possível este trabalho e que fizeram parte direta ou indiretamente deste trabalho, seja com uma contribuição no âmbito acadêmico, ou dando um apoio em um momento oportuno.”

AGRADECIMENTOS

Alguns definem com sorte, acaso, ou a ocorrência de uma remota probabilidade. Os filósofos definem como o mundo inteligível, o mundo das ideias, onde todas as coisas podem ser contempladas. Os seguidores do cristianismo definem como Deus e, algumas outras denominações até dão outros nomes.

É a Ele, em primeiro lugar, que gostaria de agradecer. Por colocar tantas pessoas boas no meu caminho, para me auxiliar, a fim de que este trabalho pudesse ser realizado. Por me abençoar com muita força, determinação e disposição para trabalhar. Por fazer com que as probabilidades, embora mínimas de que certas coisas pudessem ocorrer, mas com um fator de peso muito grande e determinístico, a fé, os eventos se realizaram. Por me dar uma família que me suportou em todos os momentos, fáceis, difíceis, alegres, tristes, calmos e conturbados. Pelos amigos, colegas de trabalho que às vezes se tornam nossa segunda família, com os quais convivemos mais até que nossa própria família, se tornando cúmplices e contribuintes de nossas realizações, nossos sonhos, nossa carreira.

Em segundo lugar gostaria de agradecer ao meu pai, o principal elemento motivador deste trabalho. Além de ser meu exemplo, uma pessoa da qual me orgulho tanto.

Gostaria de agradecer ao meu professor orientador José Jean, pela imensa dedicação, compreensão, empenho e apoio em todos os momentos da realização deste trabalho. Mesmo quando eu não acreditava, ele acreditou por mim e me ajudou a enxergar o que estava escondido.

Gostaria de agradecer imensamente aos colegas e amigos do laboratório pela força e apoio, não só em relação ao trabalho, mas motivando e sendo parceiros em todos os momentos.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica por acreditarem neste trabalho, pelo espaço físico e pelos materiais e equipamentos disponibilizados para a realização deste projeto.

À CAPES pelo apoio financeiro, essencial para a manutenção do exercício ao longo desses dois anos.

Ao Ricardo Fortes de Miranda e ao Guilherme Ameloti, por nos receberem em seu Laboratório, cedendo parte do seu espaço para o desenvolvimento deste projeto.

MARCOS, W. P. **Modelagem de sistemas estocásticos utilizando cadeias de Markov - Estudo de caso para monitoramento de pragas em lavoura cafeeira.** 2014. 145f. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

Resumo

Ao se tratar de sistemas, no que diz respeito ao seu comportamento ou natureza, podem ser classificados em dois tipos: os sistemas determinísticos e os sistemas probabilísticos. Os sistemas determinísticos têm seu comportamento conhecido ou determinável ao longo do tempo, o que não se pode dizer dos sistemas probabilísticos. Estes são baseados em probabilidades de acontecimento e requerem uma ferramenta auxiliar para que sua análise possa ser realizada. Uma ferramenta que pode auxiliar nesta análise é a cadeia de Markov, objeto de estudo deste trabalho. Através de um problema de manejo de pragas de lavoura cafeeira, um modelo de cadeia de Markov foi proposto, segundo uma metodologia desenvolvida. Foram abordadas duas dimensões, a saber, incidência e pulverização. Os resultados de três anos consecutivos foram levantados, abordando percentuais de transição recorrente ao estado sem incidência e sem pulverização, bem como o percentual de estados sem incidência e sem pulverização. Também foram levantados os setores com incidências máximas e mínimas e as pulverizações e máximas e mínimas nos períodos. A cadeia de Markov demonstrou uma ferramenta eficiente no monitoramento de pragas e análise global do sistema, o que é essencial para o desenvolvimento da agricultura de precisão.

Palavras-chave: Cadeias de Markov, Sistemas Estocásticos, Controle de Pragas, Lavoura Cafeeira, Agricultura de precisão.

MARCOS, W. P. Modeling of stochastic systems using Markov chains - Case study using Markov chains to pests monitoring in coffee plantations. 2014. 145 f. Master thesis, Federal University of Uberlândia.

Abstract

Regarding the nature of systems behavior, it can be classified into two types: deterministic and probabilistic ones. Deterministic systems have a known behavior or determinable over time, which can not be said of probabilistics. These are based on event probabilities and require an auxiliary tool to be analysed. A tool that can assist in its analysis is the Markov chain, the object study of this work. Through a management problem coffee crop pests a Markov chain model was proposed, according to a methodology developed. We addressed two dimensions, namely, impact and spraying. The results of three consecutive years were raised by addressing percentiles of recurrent transition to the state without incidence and without spraying, as well as the percentage of states without incidence and without spraying. Also the sectors with maximum and minimum impact and high and low spraying for the time periods were raised. The Markov chain proved an effective tool in pests monitoring and the system global analysis, which is essential for the development of a precision agriculture.

Keywords: Markov chains, Stochastic Systems, Pest Control, Crop Coffee, Precision Agriculture.

LISTA DE FIGURAS

Figura 1 – Agricultura de precisão aplicada nas etapas do ciclo produtivo.....	3
Figura 2- Exemplo de matriz de transição (a) e diagrama de transição (Nogueira,2009).	12
Figura 3 - Diagrama de transição do modelo proposto por (Wilcox, 2011)	15
Figura 4 - Diagrama do modelo de Cadeia de Markov Nascimento-Desastre proposto por (Strigul, 2012)	19
Figura 5 - Diagrama dos modelos A, B, C propostos para descrever a dinâmica das etapas do povoamento florestal (STRIGUL et al, 2012).....	19
Figura 6 - Exemplo de resumo da coleta de dados para um talhão – parcial da planilha. ...	24
Figura 7 - Exemplo de registro de pulverizações - parcial da planilha.....	24
Figura 8 - Fluxograma para a metodologia de determinação das matrizes de transição	26
Figura 9 - Mapa ilustrativo da fazenda analisada	29
Figura 10 - Diagrama de dois estados da primeira solução proposta.....	30
Figura 11 - Diagrama de três estados da segunda solução proposta.....	31
Figura 12 - Diagrama de estados da cadeia de Markov definida. Regiões de alta/baixa incidência e pulverizado/não pulverizado.....	33
Figura 13 - Procedimento realizado pelo algoritmo de tratamento dos dados brutos de manejo de pragas	34
Figura 14 - Exemplo de registro de pulverização	35
Figura 15 - Exemplo de planilha booleana auxiliar do registro de pulverização.....	35
Figura 16 - Exemplo de planilha pronta para ser lida, referente à pulverização realizada. ...	36
Figura 17 - Procedimento realizado pelo algoritmo de tratamento dos dados brutos de manejo de pragas	36
Figura 18 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.137	
Figura 19 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.237	
Figura 20 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.338	
Figura 21 - Algoritmo para determinar a matriz de transição de dois estados (rotina repetida para cada área)	39
Figura 22 - Algoritmo para determinar a matriz de transição de quatro estados (rotina repetida para cada área).....	40
Figura 23 - Modelagem matriz de transição para TB01 - setor A - Praga 1 - ano 2011/2012 – dois estados	42
Figura 24 - Diagrama de transição para Praga1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de dois estados.....	42

Figura 25 - Modelagem matriz de transição para TB01 - setor A - Praga 1 - ano 2011/2012 – três estados	43
Figura 26 - Diagrama de transição para Praga1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de três estados.....	44
Figura 27 - Diagrama de transição para Praga1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de quatro estados.....	44
Figura 28 - Regiões vizinhas analisadas.....	45
Figura 29 - Ilustração TB26 detalhada	45
Figura 30 - Ilustração TB31 detalhada	46
Figura 31 - Diagrama de transição para Praga1, Talhão TB26, ano 2010/2011 – referente ao modelo de quatro estados.....	46
Figura 32 - Diagrama de transição para Praga1, Talhão TB31, ano 2010/2011 – referente ao modelo de quatro estados.....	47
Figura 33 - Diagrama de transição para Praga1, Talhão TB26, ano 2011/2012 – referente ao modelo de quatro estados.....	47
Figura 34 - Diagrama de transição para Praga1, Talhão TB31, ano 2011/2012 – referente ao modelo de quatro estados	48
Figura 35 - Diagrama de transição para Praga1, Talhão TB26, ano 2012/2013 – referente ao modelo de quatro estados	48
Figura 37 - Índices da transição t_{11} para o ano de (a)2010/2011 (b) 2011/2012 (c) 2012/2013.....	49
Figura 36 - Diagrama de transição para Praga1, Talhão TB31, ano 2012/2013 – referente ao modelo de quatro estados	49
Figura 38 - Percentuais de ocorrência dos estados 1 e 2 para (a) 2010/2011 (b) 2011/2012 (c) 2012/2013	50
Figura 39 - Percentuais de ocorrência dos estados 1 e 3 para (a) 2010/2011 (b) 2011/2012 (c) 2012/2013	50
Figura 40 - Porcentagem de transição t_{11} para cada área no ano 2010/2011 referente à praga 1	51
Figura 41 - Porcentagem de transição t_{11} para cada área no ano 2011/2012 referente à praga 1	51
Figura 42 - Porcentagem de transição t_{11} para cada área no ano 2012/2013 referente à praga 1	52
Figura 43 - Diagrama de estados da segunda solução proposta com interpretação das transições.	55
Figura 44 - Triângulo de pascal	58

LISTA DE TABELAS

Tabela 1: Resumo de classificações de estados e cadeias (Nogueira, 2009)	13
Tabela 2 - Áreas de aplicação divididas em categorias.....	21
Tabela 3 - Comparação entre os modelos da literatura.....	22
Tabela 4 – Exemplo de talhão, áreas e pragas analisadas pelo controle fitossanitário de uma fazenda.....	23
Tabela 5 - Valores percentuais críticos para a incidência das pragas	24
Tabela 6 - Áreas e pragas utilizadas na modelagem do problema	34
Tabela 7 - Setores com máximo e mínimo de ocorrência dos estados de baixa incidência para o ano 2010/2011	52
Tabela 8 - setores com máximo e mínimo de ocorrência dos estados sem pulverização para o ano 2010/2011	53
Tabela 9: COmparação entre odelo geral e média dos modelos – praga 1	53

LISTA DE ABREVIATURAS

- MDP – “Markovian Decision Problem”
- EMV – “Expected Monetary Value”
- FPTC – “First Passage Time Criteria”
- IPM – “Integrated Pest Management”
- CA – “cellular automata”
- HPMM – “Hidden Phase-type Markov Chains”
- AP – Agricultura de precisão
- INTC – Intel Corporation
- MSFT – Microsoft
- IBM – International Business Machines
- MDP – Problema de Decisão Markoviano
- PPA – Aproximação de Plasticidade Perfeita

SUMÁRIO

CAPÍTULO I - INTRODUÇÃO	1
Objetivos.....	2
Justificativa	2
Estrutura da Dissertação.....	4
CAPÍTULO II - FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA	5
2.1. Processos Estocásticos.....	5
2.2. Processo Markoviano	6
2.3. Cadeia de Markov.....	6
2.4. Transições de Probabilidade	7
2.5. Representações e Características	11
2.6. Trabalhos Relacionados	13
2.7. O Problema das Pragas.....	23
CAPÍTULO III - METODOLOGIA DO PROJETO PROPOSTO	25
3.2. Metodologia.....	26
CAPÍTULO IV - DESENVOLVIMENTO DO MODELO.....	29
4.1. Definição dos estados	30
4.1.1 Abordagem 1.....	30
4.1.2 Abordagem 2.....	31
4.1.3 Abordagem 3.....	32
4.2. Tratamento dos dados.....	33
4.3. Transformação dos dados em estados.....	36
4.4. Determinação das probabilidades de transição em montagem da matriz de transição.....	38
CAPÍTULO V - RESULTADOS	41
5.1. Exemplo de modelo segundo abordagem 1	41
5.2 Exemplo de modelo segundo abordagem 2	43
5.3 Exemplo de modelo segundo abordagem 3	44
5.4 TB 26 e Tb 31 – Apresentando regiões vizinhas	44
CAPÍTULO VI – DISCUSSÃO	54
6.1 Discussão do modelo.....	54
6.1. Discussão dos resultados.....	55
6.1.1 Comparação entre os modelos.....	55
6.1.2 Análise das regiões vizinhas	56
6.1.3 Explosão de estados	58
CAPÍTULO VII – CONCLUSÃO	59
CAPÍTULO VIII – TRABALHOS FUTUROS	61
REFERÊNCIAS BIBLIOGRÁFICAS.....	63
APÊNDICE A - CÓDIGO PARA O TRATAMENTO DOS DADOS BRUTOS.....	68
APÊNDICE B - CÓDIGO DESENVOLVIDO EM MatLab® PARA A OBTENÇÃO DAS MATRIZES DE TRANSIÇÃO	72
APÊNDICE C - MATRIZES DE TRANSIÇÃO ENCONTRADAS	80
APÊNDICE D - MODELO GERAL E MÉDIA DAS MATRIZES ENCONTRADAS	132
ANEXO A - EXEMPLO DE PLANILHA DE MANEJO ECOLÓGICO.....	134

CAPÍTULO I

INTRODUÇÃO

Em se tratando de sistemas, é difícil abordá-los sem que se faça sua modelagem ou equacionamento, ao menos parcial. Conhecendo o sistema, pode-se caracterizá-lo, realizar estudos ou análises e propor melhorias visando o bem comum e a facilitação do dia-a-dia do ser humano. Este é um dos desafios da engenharia.

Ao lidar com sistemas em geral, no que diz respeito ao seu comportamento ao longo do tempo, o engenheiro pode se deparar com dois tipos de sistemas bem distintos: os sistemas determinísticos e os sistemas probabilísticos.

O primeiro é governado por equações cujo comportamento ao longo do tempo é conhecido ou determinável. O segundo entra no amplo universo da possibilidade, da probabilidade de acontecimento.

Faz-se necessário então, o uso de ferramentas que permitam o estudo deste tipo de sistemas, os quais são muito frequentes na realidade do engenheiro.

Em se tratando da aplicabilidade dos sistemas probabilísticos, pode ser citado o problema de incidência de pragas na lavoura de café. Sendo um sistema biológico, é inegável que há diversos fatores bióticos e abióticos que interferem na produtividade e no custo do mesmo. Esses fatores, normalmente não possuem um comportamento conhecido ao longo do tempo, nem sua influência no sistema é evidente. Isto o torna difícil de ser equacionado.

Dentre os custos que são evidentes, estão o monitoramento de pragas e a aplicação de defensivos agrícolas. Este é realizado de maneira constante e é baseado na experiência do cafeicultor. O monitoramento das pragas é realizado de maneira sistemática com uso de mão de obra humana.

Um problema observado é que a prevenção/controlé é realizada apenas com base na experiência do próprio agricultor. Este trabalho poderia ser realizado com auxílio de uma ferramenta dando, assim, um embasamento para que a observação do cafeicultor possa ser sustentada. A difícil tarefa que se tem é: como prever a incidência e como este modelo pode ser integrado ao sistema de controle e monitoramento de pragas.

Uma dessas ferramentas é a cadeia de Markov. É uma ferramenta de modelagem de sistemas estocásticos, que faz uso das probabilidades de transição entre estados para estudo do comportamento de sistemas ao longo do tempo. A cadeia de Markov possibilita a extração de características que permitem a previsão e tomada de decisão a respeito destes sistemas. O objeto de estudo deste trabalho é o uso dessa ferramenta para modelagem de um sistema probabilístico com dados de um caso real.

Objetivos

O presente trabalho objetiva desenvolver o modelo em cadeia de Markov para o manejo de incidência de pragas em uma lavoura cafeeira. Possui como objetivos específicos:

- Estudar cadeias de Markov e sua utilização na modelagem de sistemas estocásticos;
- Definir modelos de cadeia de Markov para o problema apresentado;
- Desenvolver um sistema de tratamento de dados;
- Modelar matematicamente o estudo de caso real;
- Realizar a análise dos resultados da aplicação do modelo ao objeto de estudo;

Justificativa

A competitividade na agricultura tem se acentuado a cada ano, o que requer profissionais rurais cada vez mais capacitados e com uma habilidade de gestão cada vez maior. Dentro deste cenário, a agricultura de precisão (AP) tem ganhando espaço no País à medida que técnicas, tecnologia e conhecimento chegam ao produtor rural.

A agricultura de precisão é o sistema de produção adotado por agricultores de países de tecnologia avançada, denominado por *Precision Agriculture*, *Precision Farming* ou *Site-Specific Crop Management* de acordo com (Nunes, 2014). Ela surgiu como um sistema de gerenciamento de informações e teve seu crescimento potencializado a partir de avanços da tecnologia de referenciamento e posicionamento, como o GPS (*Global Positioning System*) e de tecnologias de sensoriamento remoto.

A ideia é migrar de uma análise homogênea, com conceito da necessidade média para a aplicação dos insumos (fertilizantes, defensivos, água, dentre outros), para uma filosofia de gerenciamento agrícola que parte de informações confiáveis, e se completa com

decisões exatas em todo o ciclo produtivo, atuando desde a preparação do solo até a realização da colheita, ver Figura 1. O ciclo da safra pode ser entendido pelas etapas de preparo do solo, plantio, acompanhamento da lavoura e colheita. A preparação do solo refere-se à aplicação de fertilizantes e corretivos com taxas variáveis, dependendo de cada área. No plantio, as taxas podem ser variáveis, de acordo com o potencial produtivo de cada área. O acompanhamento da lavoura consiste no mapeamento de pragas e doenças, e na aplicação de defensivos agrícolas de maneira localizada. A colheita, realizada com máquinas com sensores de produtividade, fecha o ciclo com a geração de mapas de produtividade e a preparação do solo recomeça.

Assim, o campo produtivo é gerido metro a metro, levando em conta o fato de que cada pedaço da fazenda tem propriedades diferentes. O principal conceito é aplicar os insumos no local correto, no momento adequado, as quantidades de insumos necessários à produção agrícola, para áreas cada vez menores e mais homogêneas, tanto quanto a tecnologia e os custos envolvidos o permitam.

Esse trabalho se justifica pela importância acadêmica/mercado: disponibilizar um modelo eficiente cujos resultados podem ser aplicados na prática.



Figura 1 – Agricultura de precisão aplicada nas etapas do ciclo produtivo

Fonte: <http://www.agrolink.com.br/georreferenciamento/AgriculturaPrecisao.aspx> - acessado em 18/12/2014 8h30min

Estrutura da Dissertação

Primeiramente, foi realizada uma fundamentação teórica sobre cadeias de Markov no Capítulo 2. Neste capítulo, conceitos de cadeias de Markov foram explicitados, bem como alguns exemplos de aplicações encontrados na literatura.

Em seguida, a metodologia utilizada para a modelagem do comportamento a respeito da incidência de pragas é apresentada no Capítulo 3.

O capítulo 4 apresenta o desenvolvimento dos modelos seguindo a metodologia definida no capítulo anterior.

Os resultados são apresentados no Capítulo 5 e discutidos no Capítulo 6.

A conclusão é feita no Capítulo 7

Enfim os trabalhos futuros são propostos no Capítulo 8.

CAPÍTULO II

FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA

2.1. Processos Estocásticos

Um processo estocástico de acordo com (Kijima,1997) é definido como uma coleção de variáveis aleatórias $X(t)$ indexadas por um parâmetro t (geralmente é o tempo) pertencente a um conjunto \mathcal{T} .

Este parâmetro é chamado um processo discreto no tempo se o conjunto de índices for $\mathcal{T} = \mathbb{Z}_+ = \{0,1,2, \dots\}$, ou um processo contínuo no tempo se $\mathcal{T} = \mathbb{R}_+ = [0, \infty)$.

Frequentemente \mathcal{T} é tomado para ser o conjunto dos inteiros não-negativos (porém outros conjuntos são perfeitamente possíveis) e $X(t)$ representa uma característica mensurável de interesse no parâmetro t .

De acordo com (Nogueira, 2009) Processos estocásticos são interessantes para descrever o procedimento de um sistema operando sobre algum período de tempo. Com isso, $X(t)$ geralmente representa o estado do sistema no parâmetro t (geralmente o tempo). Exemplificando, $X(t)$ pode ser o nível de estoque de um determinado produto no fim de um período de tempo t .

Portanto pode-se afirmar que $X(t)$ é definido em um espaço denominado espaço de Estados.

Os processos Estocásticos podem ser classificados como:

2.1.1. Em relação ao Estado

- Estado discreto (cadeia): $X(t)$ é definido sobre um conjunto enumerável ou finito.
- Estado Contínuo (sequência): $X(t)$ caso contrário

2.1.2. Em relação ao tempo:

- Tempo discreto: t é finito ou enumerável
- Tempo contínuo: t caso contrário.

2.2. Processo Markoviano

Um processo estocástico é dito ser um processo Markoviano se:

$$\begin{aligned} P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \dots, X(t_1) = x_{t_1}, X(t_0) = x_{t_0}\} \\ = P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\} \end{aligned} \quad (01)$$

Para: $t_0 \leq t_1 \leq \dots \leq t_k \leq t_{k+1} = 0, 1, \dots$ e toda sequência $k_0, k_1, \dots, k_{t-1}, k_t, k_{t+1}$.

A expressão acima pode ser traduzida por: a probabilidade condicional de qualquer evento futuro, dado qualquer evento passado e o estado presente $X(t_k) = x(t_k)$, é independente do evento passado e depende somente do estado presente. Em termos mais resumidos, um processo estocástico é dito ser um processo Markoviano se o estado futuro depende apenas do estado presente e não dos estados passados. Esta é a propriedade de Markov, a qual é provada por (Norris, 1997).

Este tipo de processo estocástico também é denominado de *memoryless process*, uma vez que o passado é “esquecido”.

As probabilidades condicionais $P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\}$ são denominadas probabilidades de transição e, representam, portanto a probabilidade do estado $X(t_{k+1})$ ser x_{k+1} no instante t_{k+1} dado que o estado $X(t_k)$ é x_k no tempo t_k .

2.3. Cadeia de Markov

Um processo Markoviano é dito ser uma cadeia de Markov quando as variáveis randômicas $X(t)$ estão definidas em um espaço de Estados discreto E .

Quando o tempo é discreto, a cadeia de Markov é dita ser uma cadeia de Markov em tempo discreto. Neste caso, tem-se:

$$\begin{aligned} P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \dots, X(t_1) = x_{t_1}, X(t_0) = x_{t_0}\} \\ = P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\} \end{aligned} \quad (02)$$

∀ sequência $0, 1, \dots, k-1, k, k+1$.

As probabilidades de transição $P\{X(k+1) \leq x_{k+1} | X(k) = x_k\}$ representam, portanto, a probabilidade do estado $X(k+1)$ ser x_{k+1} no tempo $k+1$ dado que o estado $X(k)$ é x_k no tempo k .

Se para cada x_{k+1} e x_k , tem-se:

$$P\{X(k+1) \leq x_{k+1} | X(k) = x_k\} = P\{X(1) \leq x_1 | X(0) = x_0\} \quad (03)$$

\forall sequência $0, 1, \dots, k-1, k, k+1$.

Então, as probabilidades de transição são ditas estacionárias. Assim, tendo-se probabilidades de transição estacionárias, implica que as probabilidades de transição não mudam em relação ao tempo. Ainda de acordo com a expressão acima, as probabilidades de transição são denominadas probabilidades de transição de passo 1.

2.4. Transições de Probabilidade

Seja o espaço de estado \mathcal{N} finito ou contável infinito definido por:

$$p_{ij}(n, n+1) = P[X_{n+1} = j | X_n = i], n = 0, 1, \dots \quad (04)$$

A probabilidade condicional $p_{ij}(n, n+1)$ é a probabilidade de transição de passo 1 do estado i para o estado j no intervalo de tempo. A matriz:

$$\mathbf{P}(n, n+1) = (p_{ij}(n, n+1)) \quad (05)$$

É a matriz de transição (de passo unitário) no tempo n . De acordo com a propriedade de Markov, a probabilidade de transição é de fato igual a:

$$p_{ij}(n, n+1) = P[X_{n+1} = j | X_0, \dots, X_{n-1}, X_n = i] \quad (06)$$

E as probabilidades de transição de passo m no tempo n são definidas por:

$$p_{ij}(n, n+m) = P[X_{n+m} = j | X_n = i], n = 0, 1, \dots \quad (07)$$

E a correspondente matriz de passo m no tempo n é:

$$\mathbf{P}(n, n + m) = (p_{ij}(n, n + m)) \quad (08)$$

Particularmente, escreve-se:

$$p_{ij}(m) = p_{ij}(0, m); \quad \mathbf{P}(m) = \mathbf{P}(0, m) \quad (09)$$

Ou seja, as probabilidades de transição de passo m e a matriz de transição, respectivamente, no tempo 0.

As matrizes de transição podem não ser quadradas. As matrizes de transição de passo 1 podem ser usadas para determinar as matrizes de transição de passo m . Veja:

$$\begin{aligned} p_{ij}(n, n + 2) &= P[X_{n+2} = j | X_n = i] \\ &= \sum_k P[X_{n+1} = k, X_{n+2} = j | X_n = i] \\ &= \sum_k P[X_{n+1} = k | X_n = i] P[X_{n+2} = j | X_n = i, X_{n+1} = k] \end{aligned} \quad (10)$$

Onde a soma é tomada sobre todas as possibilidades de X_{n+1} . A terceira igualdade é devido a regra da cadeia das probabilidades condicionais. Contudo, pela propriedade de Markov tem-se que:

$$p_{ij}(n, n + 2) = \sum_k p_{ik}(n, n + 1) p_{kj}(n + 1, n + 2), \quad (11)$$

$$n = 0, 1, \dots$$

As equações acima podem ser escritas na forma matricial como:

$$\mathbf{P}(n, n + 2) = \mathbf{P}(n, n + 1) \mathbf{P}(n + 1, n + 2) \quad (12)$$

Independentemente da existência ou não as matrizes de transição são quadradas. Além disso, por um argumento de indução, segue-se que:

$$\mathbf{P}(n, n + m) = \mathbf{P}(n, n + 1) \mathbf{P}(n + 1, n + 2) \dots \mathbf{P}(n + m - 1, n + m) \quad (13)$$

Para todo $m = 0, 1, 2, \dots$

Note que $\mathbf{P}(n, n) = \mathbf{I}$, a matriz identidade, para todo n . Note também que, desde que o produto matricial não é comutativo, de duas matrizes em geral, a ordem das matrizes de transição da equação acima é extremamente importante.

Da equação 13 pode se ver que:

$$\mathbf{P}(m, n) = \mathbf{P}(m, l)\mathbf{P}(l, n) \quad m \leq l \leq n, \quad (14)$$

Ou de maneira equivalente:

$$p_{ij}(m, n) = \sum_k p_{ik}(m, l)p_{kj}(l, n), \quad m \leq l \leq n \quad (15)$$

Esta equação é conhecida como equação de Chapman-Kolmogorov (Kijima, 1997).

Definição: Uma matriz $\mathbf{A} = a_{ij}$ é dita ser estocástica se

$$a_{ij} \geq 0 \wedge \sum_j a_{ij} = 1 \quad \forall i, j \quad (16)$$

Neste caso, escrevemos $\mathbf{A} \in \mathcal{S}$, *i. e.*, \mathcal{S} denota a classe das matrizes estocásticas.

Agora, seja $\alpha_i \equiv P[X_0 = i]$ e $\boldsymbol{\alpha} = (\alpha_i)$. O vetor coluna $\boldsymbol{\alpha}$ é chamado de distribuição inicial de $\{X_n\}$. Note que o vetor probabilidade $\boldsymbol{\alpha}$ no sentido de $\alpha_i \geq 0$ e $\sum_i \alpha_i = 1$. Definindo as probabilidades incondicionais por:

$$\pi_i(n) = P[X_n = i], \quad n = 0, 1, \dots, \quad (17)$$

O vetor coluna $\boldsymbol{\pi}(n) = (\pi_i(n))$ é chamado de distribuição do estado de $\{X_n\}$ no tempo n . É claro $\boldsymbol{\pi}(0) = \boldsymbol{\alpha}$, a distribuição inicial.

Da equação de Chapman-Kolmogorov, temos que:

$$\boldsymbol{\pi}^T(n+1) = \boldsymbol{\pi}^T(n)\mathbf{P}(n, n+1), \quad n = 0, 1, \dots \quad (18)$$

Segue que a distribuição do estado pode ser computada por:

$$\boldsymbol{\pi}^T(n) = \boldsymbol{\alpha}^T \mathbf{P}(n), \quad n = 0, 1, \dots \quad (19)$$

Note que, desde que $P(n)$ seja estocástica, $\pi(n) \geq 0$ e $\pi^T(n)1 = 1$ da equação (19), da qual $\pi(n) \in \mathcal{P}$ para todo, $n = 0, 1, \dots$

Se as probabilidades de transição de passo unitário são independentes do tempo n , a cadeia de Markov $\{X_n\}$ é dita temporalmente homogênea, ou simplesmente homogênea.

Assumindo que $\{X_n\}$ é homogênea pode ser definido:

$$p_{ij} \equiv P[X_{n+1} = j | X_n = i] \quad n = 0, 1, \dots \quad (20)$$

Conforme equação (04). Para uma cadeia de Markov homogênea, a matriz de transição não depende do tempo e podemos indicar por $\mathbf{P} = p_{ij}$. Ou ainda,

$$\mathbf{P} = \mathbf{P}(n, n + 1), \quad n = 0, 1, \dots \quad (21)$$

Note que a matriz de transição \mathbf{P} é necessariamente quadrada e os possíveis estados de cada X_n são os mesmos. No que segue, vamos tratar apenas cadeias de Markov homogêneas.

De uma cadeia de Markov $\{X_n\}$ com espaço de estado \mathcal{N} , a probabilidade condicional do evento A dado $X_0 = i$ é denotada por:

$$P_i[A] = P[X_0 = i], \quad i \in \mathcal{N} \quad (22)$$

E o operador correspondente a i Então, devido à homogeneidade, a probabilidade de transição do estado i para o estado j é escrita como:

$$p_{ij} = P_i[X_1 = j] = P[X_{n+1} | X_n = i]. \quad (23)$$

Quando a distribuição inicial de $\{X_n\}$ e $\alpha \in \mathcal{P}$, a probabilidade condicional do evento A é denotada por $P_\alpha[A]$ e o operador correspondente por E_α . A probabilidade do estado é dada por:

$$\pi_i(n) = P_\alpha[X_n = i] \quad (24)$$

A matriz de transição de passo m na equação (13) torna-se:

$$\mathbf{P}(n, n + m) = \mathbf{P}^m, \quad m = 0, 1, \dots, \quad (25)$$

Isto é, o m -ésimo poder de \mathbf{P} , o qual é independente de n . A probabilidade de transição de passo m no tempo n é, todavia independente de n e é o (i, j) -ésimo componente de \mathbf{P}^m . Pode-se escrever:

$$p_{ij}(m) = P_i[X_m = j] = P[X_{m+n} = j | X_n = i] \quad (26)$$

Conforme equação (07).

A equação de Chapman-Kolmogorov Eq. (15) pode ser expressa como:

$$p_{ij}(m+n) = \sum_k p_{ik}(m)p_{kj}(n) \quad (27)$$

Ou na forma matricial:

$$\mathbf{P}^{m+n} = \mathbf{P}^m \mathbf{P}^n \quad (28)$$

2.5. Representações e Características

As probabilidades de transição podem ser representadas por meio de uma matriz de transição. Cada posição da referida matriz a_{ij} representando a probabilidade de um estado i transitar para o estado j . Para os elementos da diagonal principal, representa a probabilidade de o estado não se modificar.

$$\begin{bmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \dots & \dots & \dots & \dots \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{bmatrix}$$

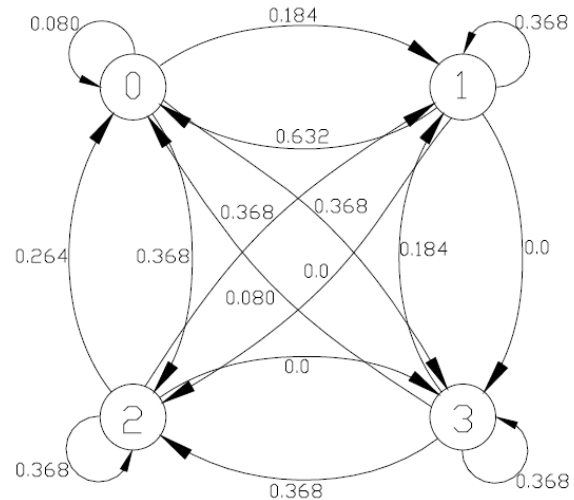
Ou ainda pode ser representado na forma de diagrama conforme ilustrado na Figura 2.

Os estados podem ser classificados e categorizados de acordo com o seu comportamento na cadeia de Markov.

Um estado j é alcançável a partir de um estado i se $p_{ij}^{(n)} > 0$ para algum $n > 0$. Isto implica que é possível o sistema entrar no estado j eventualmente quando este começa no estado i . Um estado j é dito comunicante com o estado i se o estado j é alcançável a partir do estado i e o estado i é alcançável a partir do estado j .

$$P = \begin{bmatrix} 0.080 & 0.184 & 0.368 & 0.368 \\ 0.632 & 0.368 & 0 & 0 \\ 0.264 & 0.368 & 0.368 & 0 \\ 0.080 & 0.184 & 0.368 & 0.368 \end{bmatrix}$$

(a)



(b)

Figura 2- Exemplo de matriz de transição (a) e diagrama de transição (Nogueira,2009).

Ainda se um estado i comunica-se com um estado k e este se comunica com um estado j , então i comunica-se com j .

Se dois estados comunicam entre si eles pertencem à mesma classe. Se todos os estados são comunicantes, portanto todos os estados pertencem a uma única classe, a cadeia de Markov é irredutível. Na figura 2, o estado 3 não é alcançável a partir do estado 2 pois $p_{23}^{(1)} = 0$. Todavia, este se comunica com o estado 3 através dos estados 1 e 2. Como todos os estados são alcançáveis, pode-se dizer que é uma cadeia irredutível.

Um estado é transiente se, e somente se, existe um estado j ($j \neq i$) que é alcançável a partir de i e não vice-versa, ou seja, ele não retorna ao estado i . Conseqüentemente, um estado transiente, será visitado finitas vezes.

Um estado recorrente é de tal maneira que, ao sair em determinado estado, definitivamente o processo irá retornar a este estado. Um estado é recorrente se, e somente se, não é transiente.

Há ainda o estado absorvente. Um estado absorvente se $p_{ii} = 1$. Uma vez entrando neste estado, o processo nunca deixará o mesmo. Em uma cadeia, um conjunto é dito fechado se o processo, ao entrar em um destes estados C , irá permanecer nos mesmos indefinidamente. Isto significa que nenhum estado fora de C é alcançável a partir de qualquer estado dentro de C .

Um estado i é periódico com período T se um retorno a este estado é possível somente em $t, 2t, 3t, \dots$ passos para $t > 1$ e t é o maior inteiro com esta propriedade (máximo divisor comum). Isto implica que $p_{ii}^{(n)} = 0$ sempre que n não é divisível por t .

Se há dois números consecutivos s e $s + 1$ tal que o processo pode estar no estado i nos tempos s e $s + 1$, o estado tem período 1 e é aperiódico.

Em uma cadeia de Markov de estado finito, estados recorrentes aperiódicos são chamados ergódicos. Uma cadeia é dita ergódica se todos os estados são ergódicos.

A Tabela 1 a seguir apresenta um resumo de algumas características:

Tabela 1: Resumo de classificações de estados e cadeias (Nogueira, 2009)

Conjunto Fechado	Nenhum estado, a não ser algum pertencente ao conjunto, pode ser alcançado de qualquer estado pertencente ao conjunto.
Estado Absorvente	Uma vez que se entra neste estado, nunca mais o deixa.
Estado Recorrente	Uma vez que se entra neste estado, um eventual retorno é assegurado.
Estado Periódico	O estado que pode somente ser alcançado nos passos $m, 2m, 3m, \dots$, onde m é um inteiro > 1 .
Estado Transiente	Um eventual retorno ao estado não está assegurado.
Estado Ergódico	Uma vez que se entrou neste estado, um retorno ao estado é assegurado dentro de um número finito de passos, porém o estado não é periódico e pode voltar antes de qualquer passo n .
Cadeia Irredutível	Cada estado pode ser alcançado a partir de qualquer outro estado (todos os estados são comunicantes).
Cadeia Absorvente	A Cadeia contém um ou mais conjuntos fechados e o processo poderá eventualmente ser absorvido em um dos conjuntos fechados.
Cadeia Ergódica	Todos os estados são recorrentes e aperiódicos.

2.6. Trabalhos Relacionados

Vista a generalidade do conceito, e a ampla aplicação das cadeias de Markov, abrangendo desde sistemas logísticos e modelos de mercado financeiro a problemas de sistemas biológicos e de logística, diversos e diferentes tipos de trabalhos têm sido produzidos. A seguir, alguns exemplos serão ilustrados.

(Keller Filho et al, 2006) apresentam em seu trabalho “Análise da transição entre dias secos e chuvosos por meio da cadeia de Markov de terceira ordem”, uma análise da dependência da sequência de dias secos ou chuvosos na ocorrência de chuva ou seca, através de uma cadeia de Markov. As probabilidades de transição, referente às cadeias de Markov por ele obtidas, foram estimadas com base em agregações sucessivas em diferentes períodos seguido de uma análise de variância para testar a igualdade das probabilidades de transição. Com as estimativas e uma interpolação de uma função

definida, as probabilidades diárias de transição foram obtidas. Com a análise de variância, foi decidido que a cadeia de terceira ordem era a mais adequada, pois as probabilidades de transição para ordens inferiores diferem entre si significativamente quando se consideram os diversos estados possíveis nos dias anteriores. Ele conclui em seu trabalho que em uma grande zona pluviométrica, a ocorrência de chuva, em cada dia, é dependente da sequência de dias secos e chuvosos nos três dias anteriores.

O emprego da cadeia de Markov pode ser encontrado também no meio biológico. (Kyriakidis, 1993) desenvolve algoritmo para simular o controle de pestes através da incidência de catástrofes uniformes. Primeiramente uma política ótima de não controle é estabelecida para uma condição suficiente. Caso esta condição falhe, um algoritmo de decisão Markoviano que gera uma sequência de políticas monótonas é aplicado. Por fim ele conclui que há uma forte evidência de convergência do algoritmo para uma política ótima dentro de uma classe de políticas estacionárias.

E em outro trabalho (Kyriakidis,1995) apresenta em seu trabalho uma modelagem matemática para o controle de uma população de uma peste através da inserção de um predador natural, utilizando o conceito de cadeias de Markov na modelagem da taxa de crescimento da população pelo processo de imigração. Então uma política ótima é definida com o objetivo de minimizar o custo médio por unidade de tempo a longo prazo.

Em um trabalho mais recente (Kyriakidis, 2011) introduz um processo de decisão Markoviano contínuo no tempo do controle ótimo de um simples e simétrico processo de emigração-imigração pela introdução de catástrofes totais. É provado que uma política estacionária particular é o custo-médio ótimo e, desde que considerando o problema, algumas condições extra nas taxas de custo e na taxa de transição. É importante ressaltar que, em seus trabalhos, (Kyriakidis, 2011) deu foco a escolha da política ótima dentro de uma classe generalizada. A modelagem da cadeia de Markov não foi o foco, apesar de utilizado, em seus trabalhos.

Na área da economia podemos encontrar o trabalho de (Wilcox et al, 2011), no qual ele realiza uma análise dos efeitos da liquidez em processos logísticos reversos, associando custos aos modelos Markovianos propostos por ele. O diagrama obtido por ele pode ser visualizado na Figura 3. Seu estudo contribui para análises relacionadas ao gerenciamento financeiro de processos logísticos reversos/retornos. Adotando uma perspectiva de liquidez base para os gerenciamentos, incorpora os contingentes associados com a operação reversa. Os benefícios, concluídos pelo autor, incluem a introdução de números aleatórios de unidades de entrada no sistema, bem como unidades aleatórias presentes nos vários estados do sistema. Por último, há uma contribuição para o planejamento do gerenciamento

de processo pela facilitação na construção dos intervalos de confiança associados com as características identificadas com as análises da cadeia de Markov.

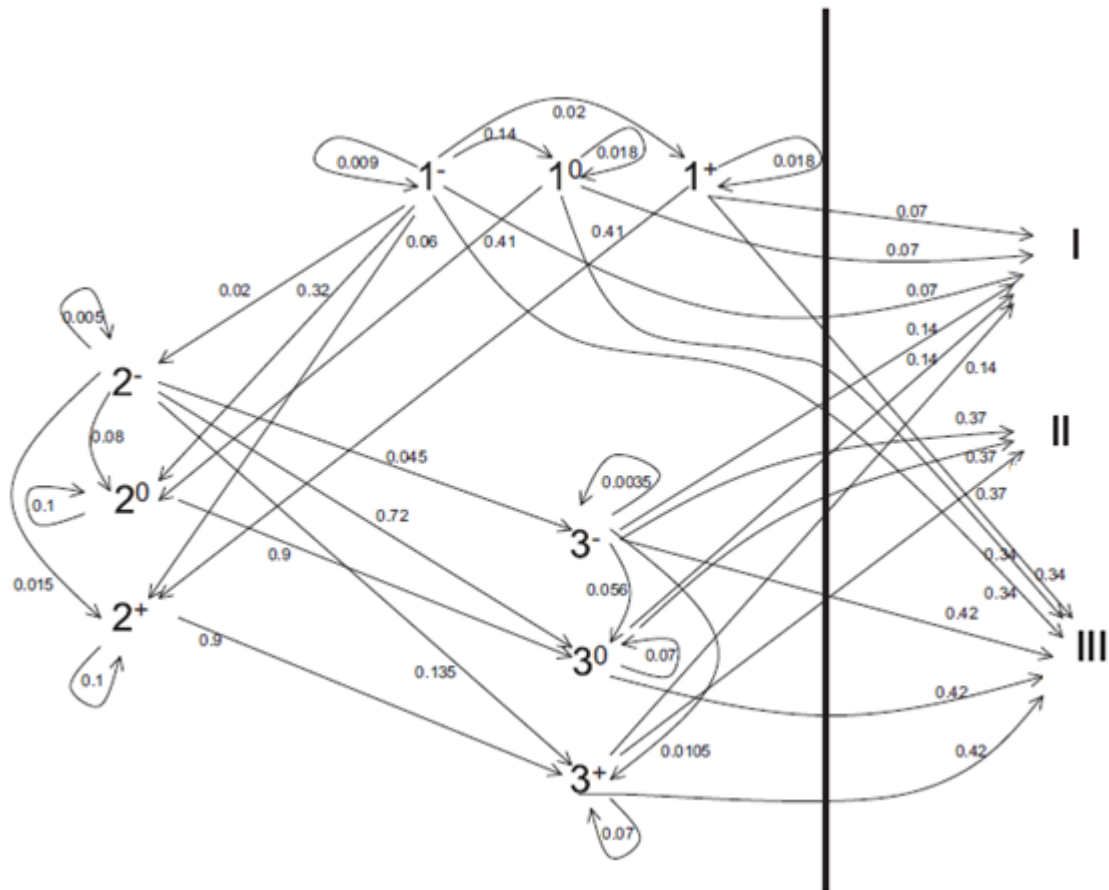


Figura 3 - Diagrama de transição do modelo proposto por (Wilcox, 2011)

Ainda no meio de gerenciamento de processos logísticos (Tilokavichai, 2012) propõe um modelo para análise de processos logísticos reversos. Utilizando um caso prático de uma empresa, ele propõe um modelo de 12 estados para possibilitar diversos tipos de análise a respeito do processo dentro desta empresa. Dentre essas análises, o autor ressalta a utilização do modelo para otimização dos custos aos quais está sujeito. Um estudo de caso é proposto a fim de validar o modelo. Os resultados encontrados demonstram método sistemático integrado que rendeu uma melhora na logística do custo efetivo no nível de satisfação do cliente. O autor finaliza ressaltando a generalidade de sua ferramenta apresentada, e sua possível aplicação em diversos outros setores.

Uma modelagem de cadeia de Markov para mercado de ações é realizada por (Masakawa, 2003). Em seu estudo o autor modela uma cadeia de Markov multivariável como um modelo estocástico da mudança de preços de portfólios. Através de análises de três portfólios (INTC, MSFT, IBM), o autor constrói um modelo multivariável utilizando o

princípio da correlação cruzada entre pares de emissões de ações. Através dos modelos obtidos aos pares, matrizes de transição referentes a quatro estados, referentes à alta e baixa dos pares (11, 1-1, -11, -1-1). O autor modela a matriz de transição com um número reduzido de parâmetros independentes. Ele ainda ressalta que para portfólios extensos o modelo resulta ineficiente, uma vez que haveria um aumento astronômico de parâmetros independentes.

(Thompson et al., 2014) desenvolvem uma simulação de perfis de densidade de massa em turfeiras boreais. Os autores apresentam um modelo de cadeia de Markov para descrever transições verticais nas propriedades das turfas de uma região do Canadá. Os objetivos deste estudo foram: determinar a distribuição de profundidade dos tipos de turfa e a densidade através de diferentes tipos de turfa; criar uma série de cadeias de Markov de estratificação de profundidade dos tipos de turfa para a simulação de perfis de turfas; e atribuir propriedades físicas aos perfis de turfas simulados utilizando cadeias de Markov. Construídas as cadeias de primeira ordem e analisadas, os autores concluem que é possível realizar simulações com o modelo por ele proposto, havendo a possibilidade de expansão do modelo para análises de tendências laterais.

(Larsen et al., 2014) utilizam cadeias de Markov para desenvolver um modelo que minimiza o transporte e os custos de armazenamento simultaneamente num problema de gerenciamento de estoque. Duas contribuições são destacadas pelos autores: Primeiramente, um modelo permitindo a flexibilidade de rotas é construído. Em segundo, foram determinadas as instâncias onde o roteamento flexível para transporte e os custos de armazenamento, os quais são significativamente menores que os casos de rota flexível. Ainda, foram encontradas reduções para casos de baixa frequência de demanda e baixo custo de transporte.

(Espada et al, 2014) modelaram políticas de redução de risco de desastres naturais utilizando problemas de decisão Markovianos - MDPs. Através da análise de diversos cenários a fim de encontrar a política ótima, os autores concluíram que os governos usam diferentes estratégias como políticas ótimas na redução de riscos. Enquanto um foca na recuperação, outro utiliza a mitigação como política ótima.

Na área de reconhecimento de imagem, (Kouritzin et al, 2014) introduziram um método de construção eficiente de probabilidades condicionais tais quais as probabilidades marginais e a covariância sobre um gráfico são mantidas nas cadeias de Markov, e este método foi usado para gerar campos aleatórios correlacionados. Diversos casos e exemplos ilustrativos foram evidenciados além de aplicações em inferência e classificação, tais como reconhecimento óptico de caracteres são empiricamente demonstrados.

(Aparana et al, 2014) desenvolveram um modelo de predição do tempo de parição de porcas a fim de reduzir a taxa de mortalidade através da supervisão planejada ou melhorando a regulação da climatização. Foram definidos os estados antes da gestação, durante a gestação e estado de imediatamente antes de entrar em trabalho de parto. Cada estado foi ainda dividido em fases. O modelo Markoviano foi usado para calcular a probabilidade de uma porca estar em determinada fase usando sensores de medição, sendo encontrados *true warnings* em 97% dos casos. Para o cálculo das probabilidades de transição foi utilizado o modelo HPMM ("*Hidden Phase-type Markov Model*"). Nesta abordagem o tempo de transição entre os estados não é fixo, logo a amostragem não é realizada em intervalos fixos de tempo.

(Arsanjani et at., 2014) desenvolveram um modelo híbrido para analisar a expansão urbana de uma área metropolitana de Teerã, Irã. O modelo híbrido consiste de um modelo de regressão logístico, cadeia de Markov e autômato celular a fim de melhorar a performance do modelo de regressão logístico tradicional. O modelo de regressão foi utilizado para criar a superfície de probabilidade e determinar os mais prováveis lugares de desenvolvimento. O modelo de Cadeia de Markov foi utilizado para quantificar a mudança. E o autômato celular foi uma ferramenta utilizada para alocar as prováveis mudanças sob condições predefinidas. Para a validação, o modelo foi avaliado por meios de valores característicos de operação relativos para diferentes conjuntos de variáveis, sendo encontrados 89% de correspondência entre modelo e vida real.

Outro trabalho analisando a expansão urbana e do uso da terra foi realizado por (Gong. et al., 2015). Utilizando sensoriamento remoto, sistemas de informação geográfica e uma combinação autômato celular – modelo de cadeia Markov foi proposto. Com tal modelo foi possível executar a projeção de trajetórias para o futuro de uma região do nordeste da China. A matriz de transição de probabilidades do problema proposto foi obtida pela análise de sobreposição espacial de dados colhidos do período entre 1989 e 2007. Então, a matriz foi utilizada na simulação das mudanças do futuro uso da terra. A validade do modelo foi obtida através da comparação dos dados simulados e observados de 2007, os quais apresentaram estatísticas em torno de 0,8. O modelo mostrou um notável crescimento na área construída e de florestas, porém uma diminuição das terras agrícolas da prefeitura de Harbin. (Luo et al., 2014) também desenvolveram um modelo de Markov para predizer as mudanças futuras no valor dos serviços ecossistêmicos de uso da terra e compara com um modelo de predição.

Ainda em relação ao uso da terra (Yang et al, 2014) utilizaram cadeias de Markov para predizer a quantificação da mudança do uso da terra, comparando-a com o método de indexação de padrões de paisagem. Foram utilizados os mapas de uso da terra de

Changping, um distrito de Beijing, no período de 1988-1998. A eficácia do modelo foi demonstrada e a precisão dos modelos foi comprovada.

(Zubizarreta et al, 2014) em seu estudo buscaram descrever a acrotonia dos ramos em um galho como uma cadeia de Markov com cinco possíveis estados correspondendo a cinco possíveis tamanhos diferentes. As matrizes de transição foram calculadas para cada galho (espirais de 3-12 ramos). Obtidas as matrizes, foi realizada uma simulação de dados estocásticos, o qual foi validado com uma base de dados independente. A validação dos resultados mostrou que a metodologia proposta reflete a variação dos tamanhos do ramo nos galhos, sugerindo a extensão da aplicação do mesmo para espécies com comportamento similar quanto ao tamanho dos ramos.

Baseado em modelo físico *indoor* e nas teorias de modelos de cadeias de Markov, (Sun et al, 2010) retrataram da migração e transformação do fenantreno no solo e nos sistemas de água subterrâneos. Dois estados foram considerados: estado de absorção (degradação fotólica, biológica, volatilização e migração para fora do lençol freático) e estado de transferência (no solo e no lençol freático). Os resultados mostraram, dentre outras coisas, que os estados de destino ambiental influenciaram a migração e a transformação de maneira diferente.

Objetivando generalizar um desenvolvimento recente da Aproximação de Plasticidade Perfeita (PPA), (Strigul et al, 2012) modelaram dinâmica de reflorestamento utilizando cadeias de Markov. Para tal, duas metodologias diferentes foram utilizadas para a construção dos modelos. A primeira construiu as probabilidades de transição baseado nos dados históricos e, posteriormente, a distribuição estacionária foi estimada. Na segunda abordagem, os dados foram utilizados para a estimativa da distribuição estacionária e, em seguida, a dinâmica da floresta é estimada. Ambos os métodos admitem a cadeia de Markov homogênea no tempo. A primeira metodologia utilizada configura um método simples de entender e muito preciso, dando uma dimensão de tempo bem definida para a matriz de transição. Em contrapartida, é dependente do intervalo de amostragem dos dados. Já na segunda, não são necessárias repetidas observações como um *snapshot* mas, é necessária de uma amostra inteira de um ano específico. A matriz de transição de probabilidade obtida representa as transições de um passo sem associação com o intervalo de tempo. Utilizando as duas metodologias, quatro modelos de matrizes de transição foram propostos. Os diagramas dos modelos propostos podem ser visualizados nas Figuras 4 e 5.

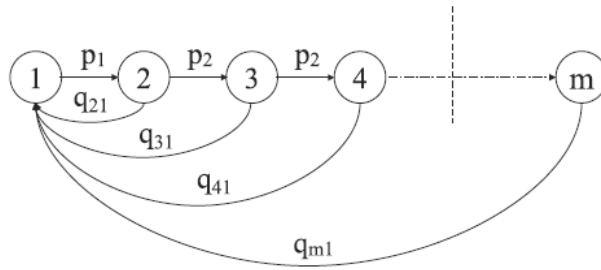


Figura 4 - Diagrama do modelo de Cadeia de Markov Nascimento-Desastre proposto por (Strigul, 2012)

Analisando os modelos obtidos, os autores concluem que a metodologia pode ser usada para outros tipos de floresta temperada e, pela análise estatística, que os parâmetros da cadeia de Markov deveriam ser específicos para aplicação em cada tipo de floresta e tipo de solo. Considerando os quatro modelos propostos, apenas o modelo C foi apropriado. As estimativas da probabilidade máxima de desastre no modelo mostraram que quatro características descrevem diferentes atributos do povoamento florestal.

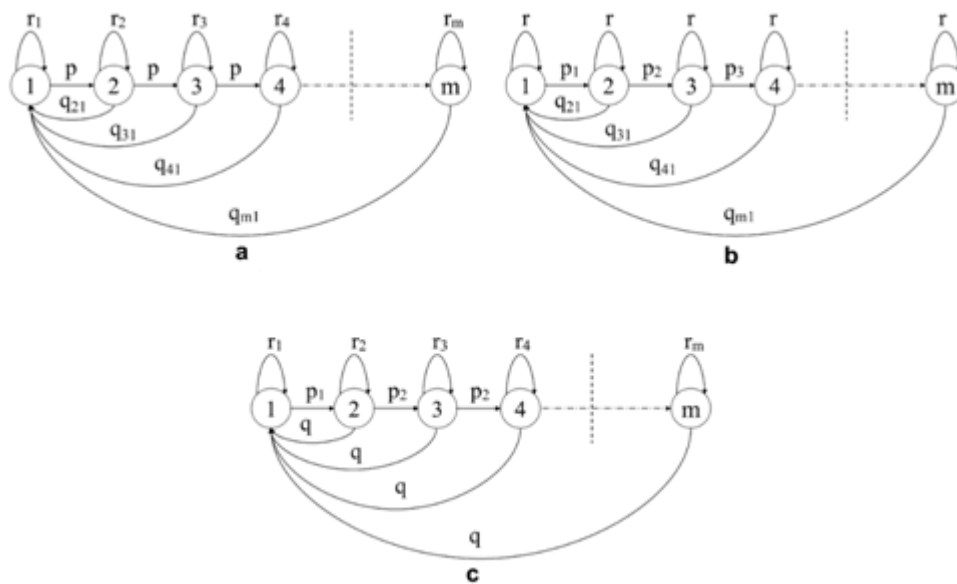


Figura 5 - Diagrama dos modelos A, B, C propostos para descrever a dinâmica das etapas do povoamento florestal (STRIGUL et al, 2012).

(Janssens et al., 2005) desenvolveram uma heurística adaptada na Modelagem de cadeias de Markov no contexto de pesquisa de transporte. Os autores assumem que cada atividade diária consiste num conjunto de sucessivas observações correlacionadas de uma variável aleatória. Esses conjuntos servem como base para a construção da cadeia segundo

a proposta de nova heurística chamada por ele de *codebooks*. Eles aplicaram a heurística na construção segundo duas abordagens: *codebooks* com maior frequência nas combinações e *codebooks* com máxima estimativa de probabilidade e compararam-nas com a construção de frequências empíricas usualmente utilizadas em cadeias de Markov de até oitava ordem. Os autores destacam como contribuições: a melhora no entendimento do comportamento dos mecanismos de organização de atividades; permite uma ferramenta de simulação para auxiliar os pesquisadores; a metodologia pode ser aplicada a outros domínios que requerem uma manipulação independente de múltiplas sequências ou que encontram dificuldades computacionais devido à explosão combinatorial no cálculo das matrizes de transição.

Muitos autores desenvolveram trabalhos teóricos a respeito de cadeias de Markov. Uma análise do tempo de primeira passagem da operação de reservatórios é realizada por (Fisher et al, 2014). Um novo algoritmo é proposto a fim de encontrar o critério de tempo de primeira passagem (FTPC) máximo ou mínimo que explora as bases da cadeia de Markov dos processos de decisão Markovianos (MDPs). O critério proposto pelos autores trabalha da mesma maneira que o valor monetário esperado (EMV) e é uma expectativa linear, logo a aproximação da solução iterativa conduz a política ótima global do FPTC. Tempo de primeira passagem esperado da uma visão no processo de decisão.

O estudo da transiência de cadeias de Markov em tempo discreto em espaços de estado gerais foi feito por (Mao et al, 2014), incluindo transiência geométrica e algébrica.

(Menshikov et al. 2014) estabeleceram teoremas gerais quantificando a noção de recorrência de cadeias de Markov contínuas no tempo irredutíveis em espaços de estado contavelmente infinitos.

(Atchadé et al, 2014) desenvolveram uma decomposição baseada nas aproximações de Martingale de uma classe geral de formas quadrática de cadeias de Markov. Para ilustração da aplicabilidade, foi discutido como a decomposição proposta pode ser usada para estudar propriedades de grandes amostras de certas estatísticas em dois problemas diferentes.

(Cheng et al, 2014) realizaram um trabalho de comparação de funções de probabilidade e análises de parâmetros de incerteza com cadeias de Markov e Monte Carlo.

(Georgiou et al, 2014) estudaram o comportamento assintótico de cadeias de Markov em passeios aleatórios não homogêneos obtendo uma fraca convergência em um *null case*. (Galtchouk et al 2014) estabelecem condições suficientes para a ergodicidade geométrica simultânea de classes de cadeias de Markov. Esta propriedade é aplicada à estimação não-paramétrica em processos de difusão ergódigos.

Uma investigação dos limites das cadeias de Markov é realizada por (Landim, 2014). (Barriga et al, 2014) desenvolveram uma análise Bayesiana de seu modelo via método de cadeia de Markov / Monte Carlo.

Como pode ser observado, diversas abordagens e análises têm sido realizadas a respeito de Cadeias de Markov e suas aplicações. No que diz respeito ao campo de aplicação, diversas áreas podem ser destacadas. A Tabela 2 a seguir possui uma análise das principais áreas relacionadas à aplicação de cadeias de Markov a partir da revisão bibliográfica realizada.

Alguns autores tentaram modelar o manejo de pragas utilizando diferentes estratégias, (Vinatier et al, 2014) utilizaram estratégias de manejo de pragas integrado (IPM). Os autores consideraram como ferramentas de eliminação de pragas a fragmentação de cultura e a captura em massa. O modelo mostrou que as armadilhas foram mais eficientes nas zonas de transição e, a redução da fragmentação das plantações também ocasionou melhora. Porém, o modelo se mostrou extremamente específico à aplicação particular tratada no trabalho.

Tabela 2 - Áreas de aplicação divididas em categorias

Área de aplicação	Quantidade	Quantidade (%)
Matemática (Análises teóricas, propriedades e teoremas)	9	31
Geografia	8	27,6
Biologia, Pecuária, Agropecuária	7	24,1
Logística	2	6,8
Economia	1	3,5
Transporte	1	3,5
Reconhecimento Imagem	1	3,5
Total	29	100

No que se refere ao modelo adotado, diferentes estratégias foram utilizadas na solução dos problemas analisados. As diferenças podem ser observadas em quesitos tais como: a ordem da matriz de transição, número de estados adotados, tipo de sistema (se é discreto ou contínuo), entre outros. Cada autor adotou a estratégia segundo seu sistema particular estudado. A Tabela 3 mostra um resumo de algumas dessas características, observadas em alguns dos trabalhos e comparadas entre si.

Quanto à definição do modelo, no que diz respeito à determinação das probabilidades de transição, cerca de 86% dos trabalhos revisados não explicitaram claramente a

metodologia utilizada para determiná-las. (Janssens D., 2005) cita em seu trabalho que a maioria dos modelos são baseados na frequência relativa das transições encontradas no histórico, propondo uma maneira alternativa de obtê-las. (Aparana et al, 2014) ainda propõem uma maneira de determinar as probabilidades de transição que não possuem um intervalo de tempo de amostragem fixo (um caso bem particular).

Tabela 3 - Comparação entre os modelos da literatura

Autor	Assunto	Tipo	Estados	Ordem	Homogeneidade	Metodologia	Definição do Modelo
Keller Filho et al, 2006	Tempo	Continua	2	1 ^a - 3 ^a	Sim	Baseado no histórico	Frequência Relativa
Tilokavichai et al, 2012	Logística	Discreta	12	1 ^a	Sim	Baseado no histórico	-
Maskawa J., 2002	Mercado de ações	Discreta	4	1 ^a	Sim	Baseado no histórico	-
Janssens D., 2005	Transporte	Discreta	5	1 ^a - 20 ^a	Sim	Baseado no histórico	Codebooks
Wilcox W. et al, 2011	Logística	Discreta	12	1 ^a	Sim	Baseado no histórico	-
Espada et al, 2014	Desastre Natural	Discreta	16	1 ^a	Sim	Baseado no histórico	-
Aparna U. et al, 2014	Pecuária	Discreta	3	1 ^a	Sim	Baseado no histórico	HPMM
Strigul et al, 2012	Dinâmica Florestal	Discreta	Até 12	1 ^a	Sim	Baseado no histórico	Mista

- definição do modelo não especificado

A ordem da cadeia proposta, também varia de trabalho a trabalho. O que se pode constatar é que cerca de 75% foram de primeira ordem, ou seja, a transição para o estado futuro foi baseado em apenas um estado para a maioria das soluções propostas. Todos os modelos trataram de cadeias homogêneas. (Apenas Keller Filho et al, 2006) trataram de cadeias contínuas. Os estados variaram entre 2 e 12.

Independente do assunto tratado, a construção dos modelos está condicionada aos dados históricos. A partir destes as probabilidades de transição foram determinadas na maioria dos casos, através da frequência relativa de ocorrência. O que faz total sentido em se tratando de modelos estocásticos, uma vez que não temos uma equação bem definida para o comportamento do modelo.

2.7. O Problema das Pragas

Uma fazenda produtora de café, geralmente está dividida em talhões. Estes talhões, por sua vez, estão subdivididos em setores. Os setores, por sua vez, são compostos por áreas. Em cada área é realizado um monitoramento periódico de pragas, mediante a coleta de amostras de folhas das plantas de acordo com o método proposto por (Bearzoti , 1994). Um exemplo de áreas e pragas pode ser visualizado na tabela 4.

O método proposto pelos autores consiste em coletar folhas nos terços médio, superior e inferior das plantas, em plantas aleatórias dentro de um talhão, sendo 20 folhas em cada terço por planta. Estas folhas são inspecionadas pelo profissional da área, o qual registra a presença ou não das pragas através da frequência relativa observada para cada uma das plantas, em cada área e para cada praga.

Os registros de cada área são realizados por setor e, a coleta dentro destes setores é realizada de maneira aleatória e bem distribuída espacialmente, conforme descrito por (Coutinho, 2005) . As coletas realizadas são organizadas. A cada período, uma planilha por talhão é gerada com os dados referentes a amostragem dos dados de pragas. Um exemplo por ser visualizado na Figura 6.

Tabela 4 – Exemplo de talhão, áreas e pragas analisadas pelo controle fitossanitário de uma fazenda

TALHÃO	Setores	Doenças	TALHÃO	Setores	Doenças
TB 01	6	8	TB 22	4	8
TB 08	5	8	TB 23	4	8
TB 09	4	8	TB 25	3	8
TB 10	7	8	TB 26	3	8
TB 11	7	8	TB 27	1	8
TB 12	2	8	TB 28	10	8
TB 13	3	8	TB 29	1	8
TB 15	2	8	TB 30	10	8
TB 16	4	8	TB 31	3	8
TB 17	4	8	TB 32	3	8
TB 18	1	8	TB 33	10	8
TB 19	17	8			
Total de setores		114			

Responsável	Período	SETOR	Praga 1	Praga 2	Praga 3	Praga 4	Praga 5	Praga 6	Praga 7	Praga 8
Indivíduo 1	1-set	A	2,0	8,7	6,8	-	3,6	-	0,6	-
Indivíduo 1	1-set	B	1,7	6,9	4,4	-	3,3	-	0,5	-
Indivíduo 2	1-set	C	2,1	7,5	3,4	-	4,1	-	0,4	-
Indivíduo 1	1-set	D	2,0	6,3	7,5	-	4,4	-	0,2	-

Figura 6 - Exemplo de resumo da coleta de dados para um talhão – parcial da planilha.

Para realizar o manejo e pragas e aplicar o insumo para o controle, é necessário reconhecer quando o índice de infestação ultrapassa certo valor limite. Os valores limites utilizados para a determinação deste, são os valores definidos pelos profissionais da área, os quais podem ser visualizados na Tabela 5. Algumas pragas são controladas a partir de certo limite. Para outras, a sua simples ocorrência na lavoura indica a necessidade de controle.

Tabela 5 - Valores percentuais críticos para a incidência das pragas

VALORES CRITICOS (%)							
Praga 1	Praga 2	Praga 3	Praga 4	Praga 5	Praga 6	Praga 7	Praga 8
5	10	5	-	-	-	-	-

Quando a alta incidência é detectada, são aplicadas as correções através de pulverizações de insumos agrícolas a fim de controlá-las. Essas aplicações são registradas em uma planilha de pulverizações, a qual está organizada em colunas que representam as quinzenas do ano e, em linhas que representam os setores de cada talhão. Um exemplo pode ser visualizado na Figura 7.

Talhão		agosto				setembro				outubro				novembro			
Talhão	Setor	1ºagosto		2ºagosto		1ºsetembro		2ºsetembro		1ºoutubro		2ºoutubro		1ºnovembro		2ºnovembro	
		Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto
1	A													10-nov	D13		
1	B													10-nov	D13		
1	C													10-nov	D13		
1	D													10-nov	D13	21-nov	F
1	E													11-nov	D13	21-nov	F
1	F													11-nov	D13	30-nov	H
8	A													2-nov	D7		
8	B													3-nov	D7		
8	C													2-nov	D7		
8	D													2-nov	D7		
8	E													2-nov	D7		

Figura 7 - Exemplo de registro de pulverizações - parcial da planilha

Conhecendo melhor o processo, pode-se então definir como obter os estados da cadeia de Markov. Para tal é necessário seguir a metodologial descrita na próxima seção.

CAPÍTULO III

METODOLOGIA DO PROJETO PROPOSTO

Realizada a revisão dos conceitos e apresentados os trabalhos relacionados a cadeias de Markov, o próximo passo é definir como obter o modelo de cadeia de Markov para a aplicação no caso de controle de pragas.

Conforme descrito no capítulo anterior, as abordagens de cadeia de Markov atingem diversas áreas e, em cada área, as soluções são bem particulares. Essa diversidade de aplicação e ampla abrangência dificultam a definição do modelo, mas ao mesmo tempo abrem os horizontes das possibilidades a serem consideradas.

Os estados foram definidos observando o sistema e a necessidade do que se pretendeu analisar, que foi: a presença ou ausência das pragas; e a presença ou ausência de aplicação de defensivos agrícolas. Os estados foram definidos segundo essas características pois, eram essas as características das quais havia uma boa base de dados.

Os estados foram ainda definidos como discretos, seguindo a abordagem realizada por (Keller filho et al, 2006) que definiu seus estados booleanos baseado na presença ou ausência de chuva.

A quantificação das matrizes de transição foi realizada através da frequência relativa de ocorrência, estratégia que é mais utilizada em se tratando de cadeias de Markov. As abordagens alternativas encontradas não foram aplicadas a este trabalho.

Os codebooks, estratégias para eliminar redundâncias, precisam ser melhor estudados antes de serem definidos na aplicação deste problema em específico. Alguns estados encontrados em grande frequência na cadeia, podem não ser redundantes e, se excluídos podem levar a uma conclusão errada sobre o comportamento do sistema.

Abordagens híbridas com tempo de amostragem variável, conforme realizado pelo HPMM também não foram utilizados devido ao método de amostragem já utilizado no manejo de pragas, que é anterior ao trabalho.

A seguir toda a metodologia utilizada neste trabalho encontra-se descrita.

3.2. Metodologia

Para a definição do modelo de cadeia de Markov deste trabalho, foi seguida a metodologia apresentada na Figura 8.

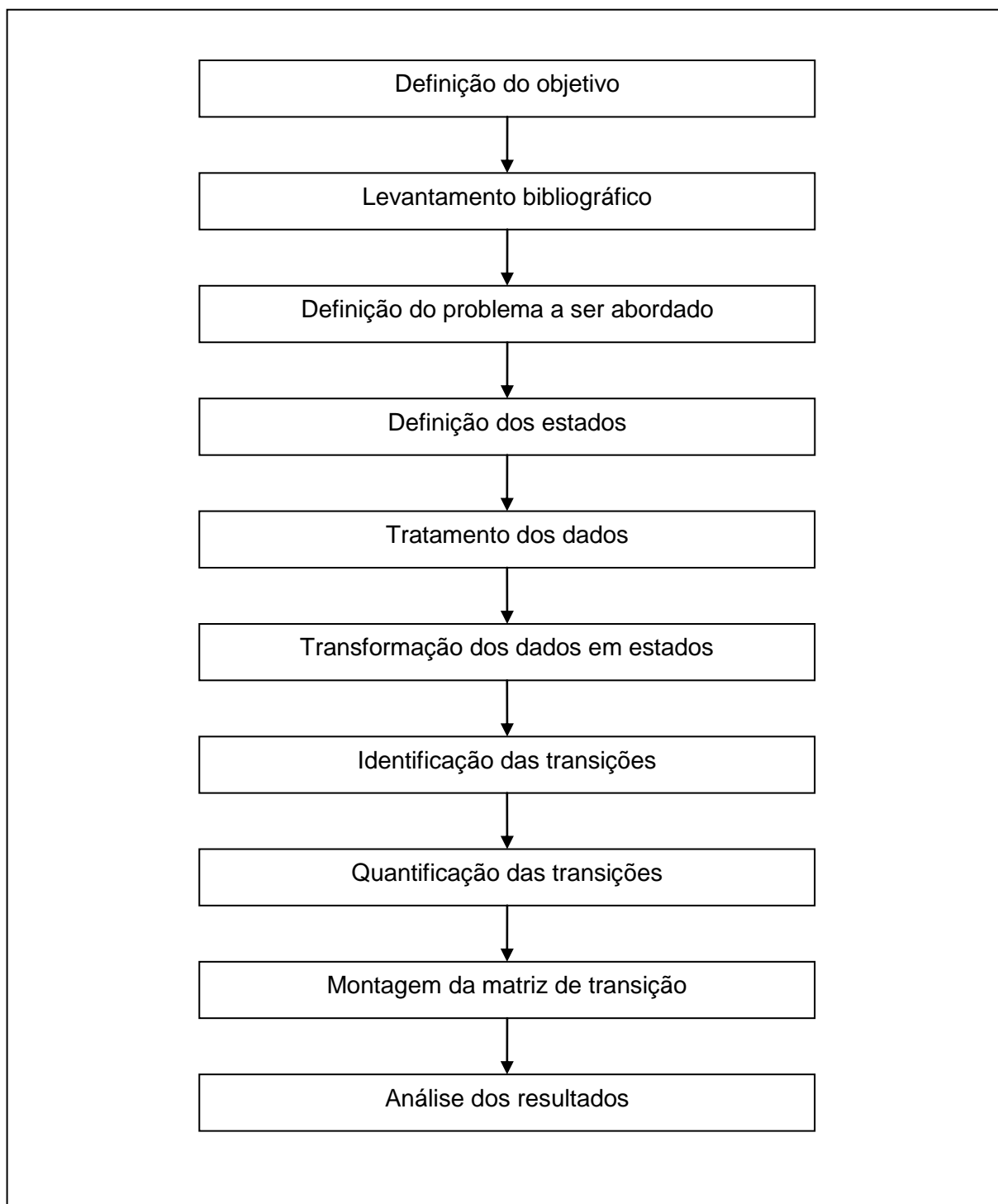


Figura 8 - Fluxograma para a metodologia de determinação das matrizes de transição

Passo 1 – Definição do objetivo: O primeiro passo deste trabalho é definir os objetivos a serem alcançados, a partir do problema encontrado. Visto o problema no manejo de pragas impreciso, sem tecnologia e com aplicações baseadas na experiência do profissional da área, uma meta a ser alcançada a fim de auxiliar no manejo dessas pragas e na aplicação dos insumos foi estabelecida.

Passo 2 – Levantamento bibliográfico: Definidas as metas, o próximo passo é realizar uma pesquisa bibliográfica para levantar os trabalhos relacionados ao problema estudado. Em seguida pode ser definida uma ou mais estratégias a serem utilizadas para resolver o problema.

Passo 3 – Definição do problema a ser abordado: Com as referências analisadas, o problema pode ser associado a uma estratégia ou solução que possa conduzir a resposta desejada. Também, neste passo já se tem uma base para definir uma estratégia que seja específica para abordar o problema.

Passo 1 - Definição dos Estados: Conhecendo a dinâmica do sistema analisado e o problema a ser resolvido, os estados são definidos baseando-se em duas características: nas propriedades observáveis do sistema, e nas informações do sistema que se deseja obter.

Passo 2 - Tratamento dos dados: Os dados brutos relativos ao sistema analisado são tratados. São retirados os dados fora dos padrões de análise ou incompletos, os quais não contribuem para o modelo a ser construído. Estes dados são dados quantitativos, referentes à frequência relativa de incidência de pragas na região analisada.

Passo 3 - Transformação dos dados em estados: Os dados quantitativos tratados no passo anterior são convertidos em estados (booleanos) neste passo

Passo 4 - Identificação da transição: Neste passo são identificadas as transições. As transições são identificadas observando-se a sequência temporal dos estados. Temos dois tipos de transições possíveis: o sistema pode permanecer no mesmo estado ou pode mudar de um estado para outro.

Passo 5 - Quantificação das transições: Identificadas as transições, o próximo passo é quantificá-las. Neste passo as transições são quantificadas através de sua frequência relativa, respeitando a propriedade de matrizes estocásticas descrita na fundamentação teórica (ver equação 16).

Passo 6 - Montagem da Matriz de Transição: Este passo se refere à organização das transições quantificadas no passo anterior.

Passo 7 – Análise dos resultados: O último passo concerne à análise das matrizes encontradas para associar às características do sistema, auxiliando no manejo de pragas e aplicação de insumos.

O modelo foi obtido através da execução dos passos descritos acima. O desenvolvimento deste, passo a passo, encontra-se no próximo capítulo.

CAPÍTULO IV

DESENVOLVIMENTO DO MODELO

Este capítulo se refere ao desenvolvimento do modelo proposto. A fazenda de lavoura cafeeira, sobre a qual este modelo está sendo realizado, está dividida em talhões, e estes por sua vez, são divididos em setores. Os setores totalizam 114 e são ainda subdivididos em áreas. Um mapa ilustrativo pode ser visualizado na Figura 9.

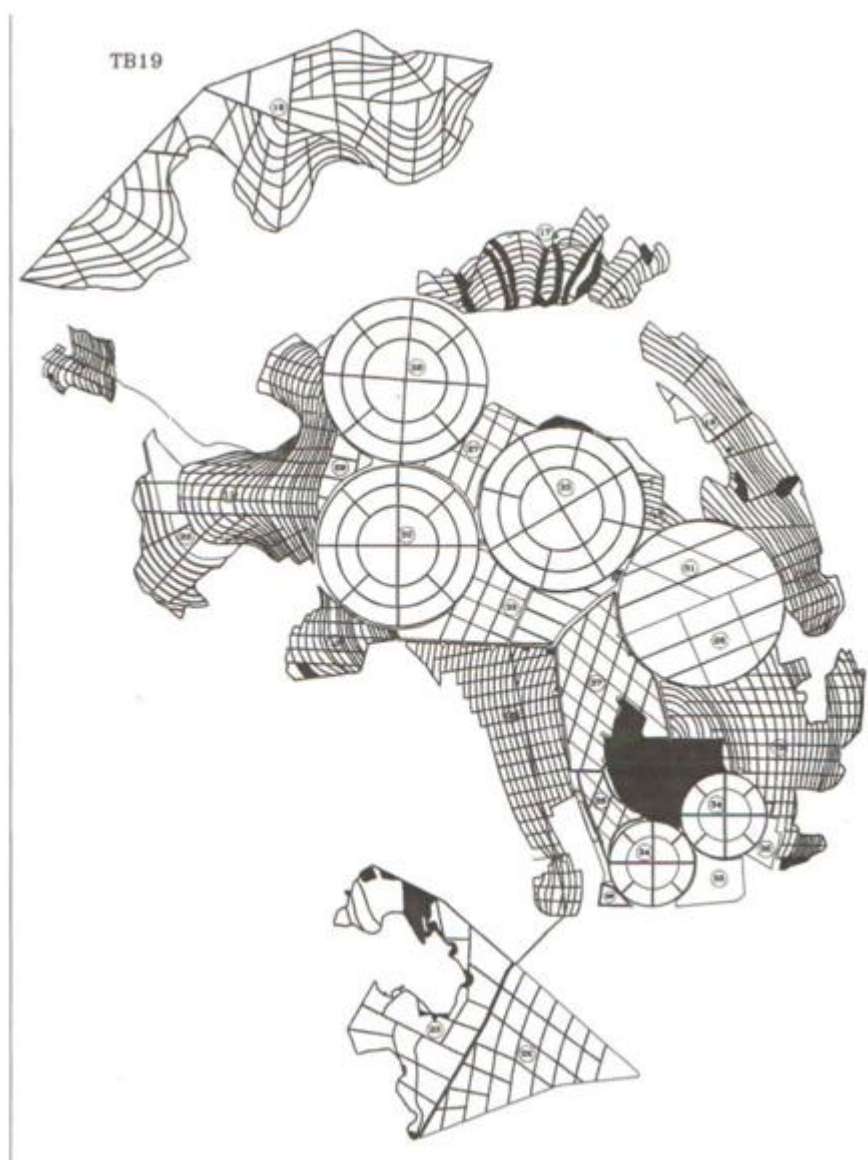


Figura 9 - Mapa ilustrativo da fazenda analisada

Nestas áreas, compostas por diversas ruas, são realizadas as coletas de amostras quinzenalmente conforme proposto por (Bearzoti,1994), e os resultados das amostras são armazenados em arquivos Excel[®]. Esses arquivos são referentes ao monitoramento de 8 pragas nos 114 setores da fazenda. E é dentro deste contexto que três abordagens distintas foram propostas neste trabalho.

4.1. Definição dos estados

4.1.1 Abordagem 1

Nesta primeira abordagem, dois estados foram considerados:

- Estado 1: para baixa incidência de pragas ;
- Estado 2: para alta incidência de pragas.

Um estado pode ser classificado baseando-se na presença ou ausência de determinada praga. Esse modelo foi assim decido baseando-se no trabalho de (Keller Filho et al.,2006).

Para que um estado fosse considerado como alta incidência, sua frequência relativa, baseada na amostragem proposta por (Bearzoti, 1994), deveria ultrapassar um valor de controle, particularmente definido por cada praga. Os autores propõem uma linha de decisão para definir qual é o valor a ser considerado, baseado em cada tipo de amostragem realizada. Porém, o valor seguido foi definido segundo os dados históricos obtidos e o valor utilizado na empresa do estudo de caso considerado (definidos na Tabela 5, anteriormente). O nome da empresa foi mantido em sigilo por questões de confidencialidade. O diagrama de transição pode ser visualizado na Figura 10.

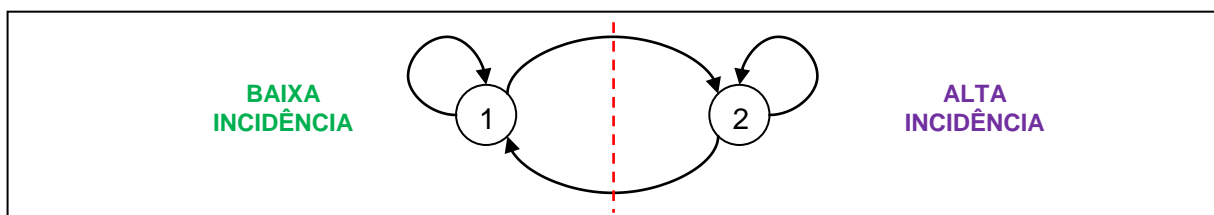


Figura 10 - Diagrama de dois estados da primeira solução proposta.

Para este estado, foi avaliada apenas a presença ou ausência da praga. Com este modelo, nada pode ser dito a respeito da aplicação de defensivos agrícolas na lavoura.

4.1.2 Abordagem 2

A segunda solução proposta foi uma cadeia de Markov de três estados:

- Estado 1: Baixa incidência de pragas;
- Estado 2: Alta Incidência de pragas;
- Estado 3: Pulverizado.

Para este modelo, o estado é definido como alta ou baixa incidência também se baseando na frequência relativa de controle definida na Tabela 5. Para o terceiro estado acrescentado, o estado pulverizado, este foi definido baseando-se na planilha pulverizações. O diagrama de estado pode ser visualizado na Figura 11.

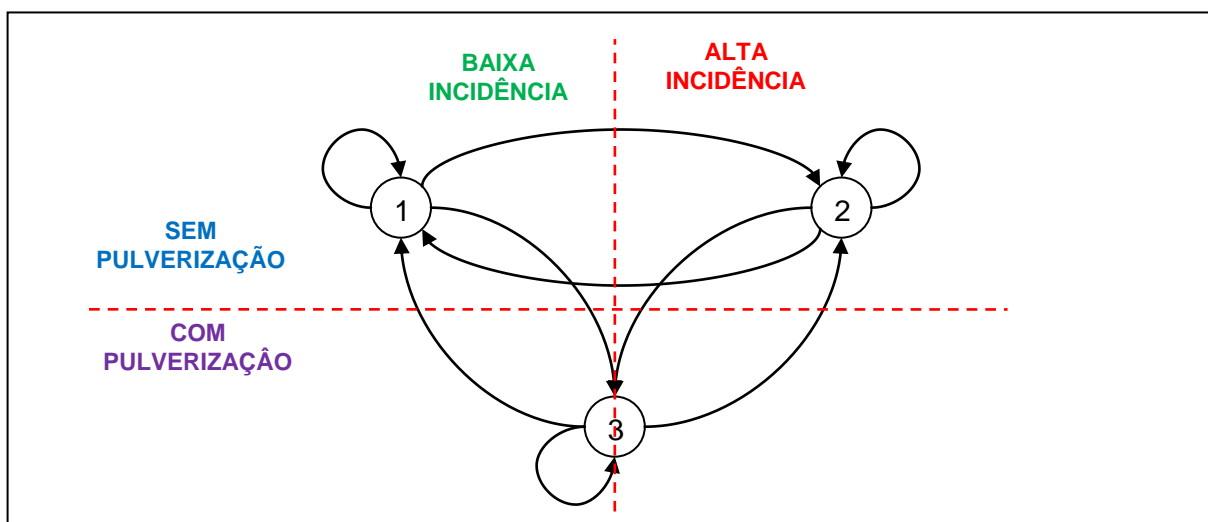


Figura 11 - Diagrama de três estados da segunda solução proposta

A planilha de dados de pragas contém um arquivo para cada quinzena, e em cada arquivo há uma planilha para cada talhão. Já planilha de pulverizações é somente uma por ano contendo o registro de todas as pulverizações realizadas em todas as áreas ao longo de todo o ano.

O estado foi definido como pulverizado, com base nesta última planilha. Havendo registro de pulverização para uma área em determinada quinzena, aquele estado foi definido como pulverizado. A manipulação dos dados está detalhada no tópico de tratamento dos dados.

4.1.3 Abordagem 3

A terceira solução proposta foi uma cadeia de Markov de quatro estados. Temos duas dimensões bem evidentes: a pulverização e a presença/ausência de pragas. Ambos são importantes na determinação dos estados e no manejo das pragas. Baseando-se nestas duas dimensões, quatro estados foram bem definidos:

- Estado 1: Baixa incidência de pragas e não-pulverizado;
- Estado 2: Alta incidência de pragas e não-pulverizado;
- Estado 3: Baixa incidência de pragas e pulverizado;
- Estado 4: Alta incidência de pragas e pulverizado.

Uma ilustração destes quatro estados pode ser visualizada no diagrama da Figura 12. Os estados foram representados e divididos por áreas.

O primeiro estado (baixa incidência e não pulverizado) é o estado que se pretende estar a maior parte do tempo. É o estado em que a planta está com o nível de pragas controlado e não há a aplicação de defensivos agrícolas.

O segundo estado (alta incidência e não pulverizado) é o estado de infestação de pestes que requer a ação de controle sobre o sistema. Quando este estado é encontrado, imediatamente deve ser realizada a pulverização a fim de retornar o estado 1, na melhor hipótese.

O terceiro estado (baixa incidência de pragas e pulverizado) é o estado que pode ser classificado como o estado de manutenção preventiva ou erro de aplicação. Embora possa ocorrer, o que se pretende é que seja o mínimo possível. Ainda é importante observar que, após a sua ocorrência, não se encontre um índice e alta infestação de pragas.

O quarto e último estado é o estado de alta incidência e pulverizado. Este é o estado que, detectado o nível de necessidade de controle da praga, é realizada a aplicação do defensivo agrícola. Este estado, embora não desejado, é um estado que pode ser transiente entre os estado 2 (alta incidência e não pulverizado) e 1 (baixa incidência e não pulverizado), ou identifica defensivos de baixa eficiência ou pragas que adquiriram resistência a determinados princípios ativos.

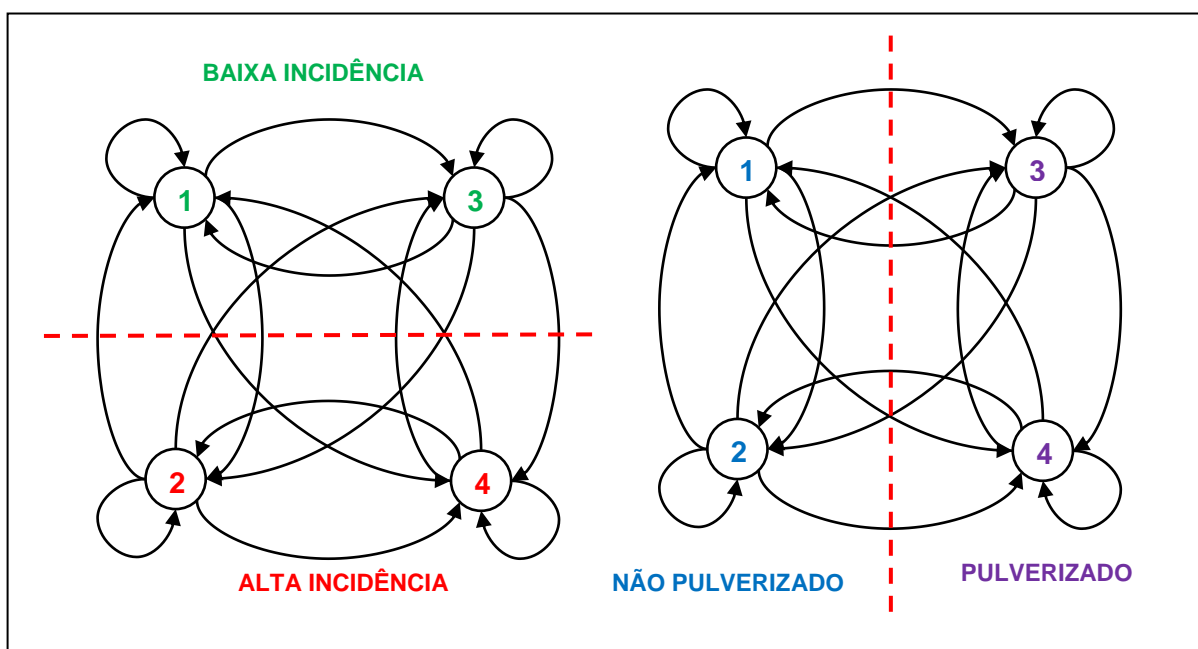


Figura 12 - Diagrama de estados da cadeia de Markov definida. Regiões de alta/baixa incidência e pulverizado/não pulverizado

4.2. Tratamento dos dados

Definido o modelo, o próximo passo é realizar o tratamento dos dados, a fim de deixá-los prontos para a modelagem. Conforme descrito no início deste capítulo, a fazenda foi subdividida em 114 setores, dos quais os dados foram dispostos. Os dados disponíveis correspondem ao controle realizado nos períodos de 2008 a 2013.

Analisando os dados, é visto que algumas áreas estão com os dados incompletos, logo elas foram descartadas da análise do problema. Há ainda o problema de duas quinzenas no ano (em abril e em dezembro) quando não é realizado o controle de pragas. Assim sendo, os dados selecionados para a modelagem correspondem a 106 áreas da fazenda, correspondendo aos períodos de agosto de 2010 a julho de 2013 para três pragas (sendo 22 quinzenas para 2010/2011, 23 quinzenas para 2011/2012 e 22 quinzenas para 2012/2013). Um resumo das áreas selecionadas pode ser visualizado na tabela 6.

Para realizar o tratamento destes dados foi utilizado o software MatLab®. Utilizando o software, um algoritmo de tratamento de dados foi desenvolvido para realizar a leitura dos dados, eliminar as áreas que dispunham de dados incompletos e salvar os dados em um arquivo.

Um fluxograma com a rotina implementada neste algoritmo pode ser visualizada na Figura 13. A matriz de dados é inicializada e a fonte dos dados brutos é identificada por ano. A matriz de dados tratados é então preenchida com esses dados brutos e é realizada a verificação se há dados vazios, os quais são eliminados da matriz. Elimina-se então os

dados referentes às pragas que não serão analisadas, obtendo a matriz com os dados tratados. O código do algoritmo de tratamento dos dados desenvolvido pode ser visualizado no apêndice A deste trabalho.

Tabela 6 - Áreas e pragas utilizadas na modelagem do problema

TALHÃO	Áreas	Doenças	TALHÃO	Áreas	Doenças
TB 01	6	3	TB 22	4	3
TB 08	5	3	TB 23	4	3
TB 09	4	3	TB 25	3	3
TB 10	7	3	TB 26	3	3
TB 11	7	3	TB 27	1	3
TB 12	2	3	TB 28	10	3
TB 13	2	3	TB 29	1	3
TB 15	2	3	TB 30	10	3
TB 16	4	3	TB 31	3	3
TB 17	4	3	TB 32	2	3
TB 18	1	3	TB 33	5	3
TB 19	16	3			
Total de áreas		106			

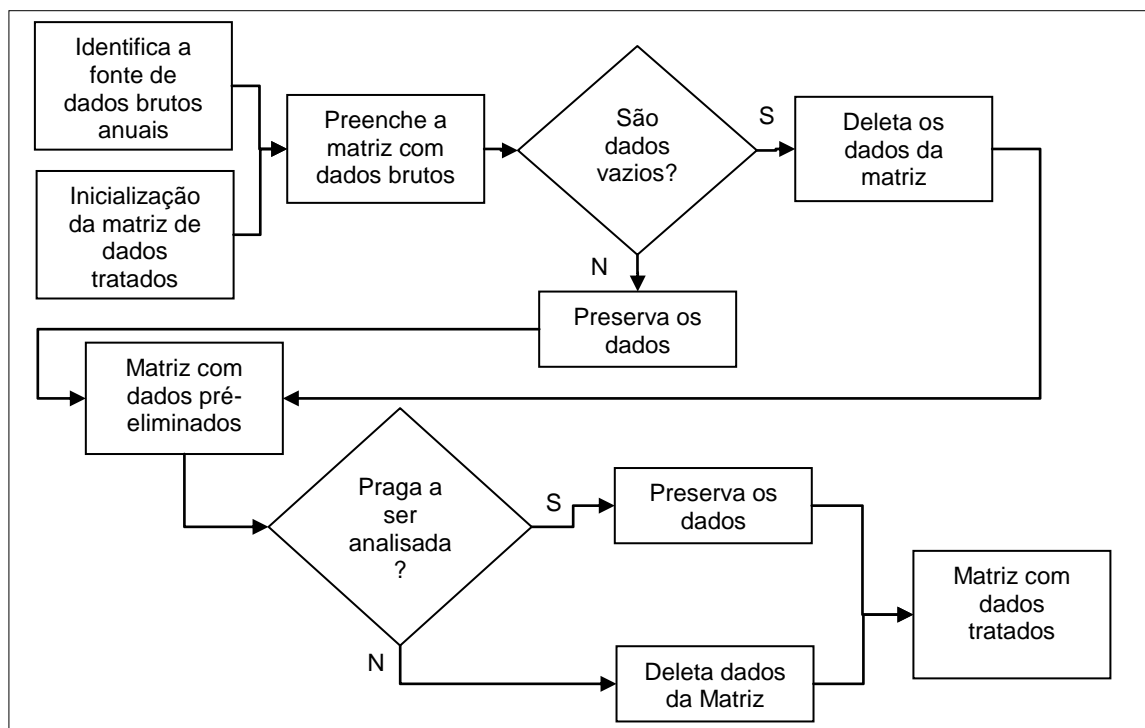


Figura 13 - Procedimento realizado pelo algoritmo de tratamento dos dados brutos de manejo de pragas

Para a planilha de pulverizações o tratamento dos dados foi realizado em duas etapas: uma em Excel® para preparar os dados a serem lidos; e outra em MatLab® para realizar a leitura dos dados e salvá-los.

A partir da planilha em Excel® existente (ver exemplo Figura 14), foi criada uma planilha auxiliar de dados booleanos indicando 1 para pulverização realizada e 0 para pulverização não realizada naquele período, conforme ilustrado na Figura 15. Em seguida uma segunda planilha foi criada para organizar os dados para leitura, conforme Figura 16.

Talhão		agosto				setembro				outubro				novembro			
Talhão	Setor	1ºagosto		2ºagosto		1ºsetembro		2ºsetembro		1ºoutubro		2ºoutubro		1ºnovembro		2ºnovembro	
		Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto
1	A													10-nov	D13		
1	B													10-nov	D13		
1	C													10-nov	D13		
1	D													10-nov	D13	21-nov	F
1	E													11-nov	D13	21-nov	F
1	F													11-nov	D13	30-nov	H
8	A													2-nov	D7		
8	B													3-nov	D7		
8	C													2-nov	D7		
8	D													2-nov	D7		
8	E													2-nov	D7		

Figura 14 - Exemplo de registro de pulverização

Talhão		agosto				setembro				outubro				novembro			
Talhão	Setor	1ºagosto		2ºagosto		1ºsetembro		2ºsetembro		1ºoutubro		2ºoutubro		1ºnovembro		2ºnovembro	
		Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto	Data	Produto
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Figura 15 - Exemplo de planilha booleana auxiliar do registro de pulverização

O algoritmo para leitura das planilhas de pulverização foi desenvolvido e implementado em MatLab®, o qual realizou as leituras da terceira planilha de cada arquivo, organizando-as em uma matriz de pulverizações. Esta matriz foi utilizada para a determinação dos estados onde houve pulverização. Este algoritmo pode ser visualizado na Figura 17. Ele identifica a fonte de dados para a pulverização por ano de realização. Em seguida verifica as regiões que serão analisadas, apagando as linhas da matriz referentes às regiões descartadas. Apaga-se, da matriz obtida, as colunas referentes às quinzenas que não serão analisadas.

A matriz obtida com a realização deste procedimento foi organizada em linhas representando cada área na qual foi aplicado o defensivo e colunas representando cada quinzena de aplicação. Ela foi utilizada para determinação dos estados de pulverização.

01/ago	ago/02	set/01	set/02	out/01	out/02	nov/01	nov/02
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0

Figura 16 - Exemplo de planilha pronta para ser lida, referente à pulverização realizada.

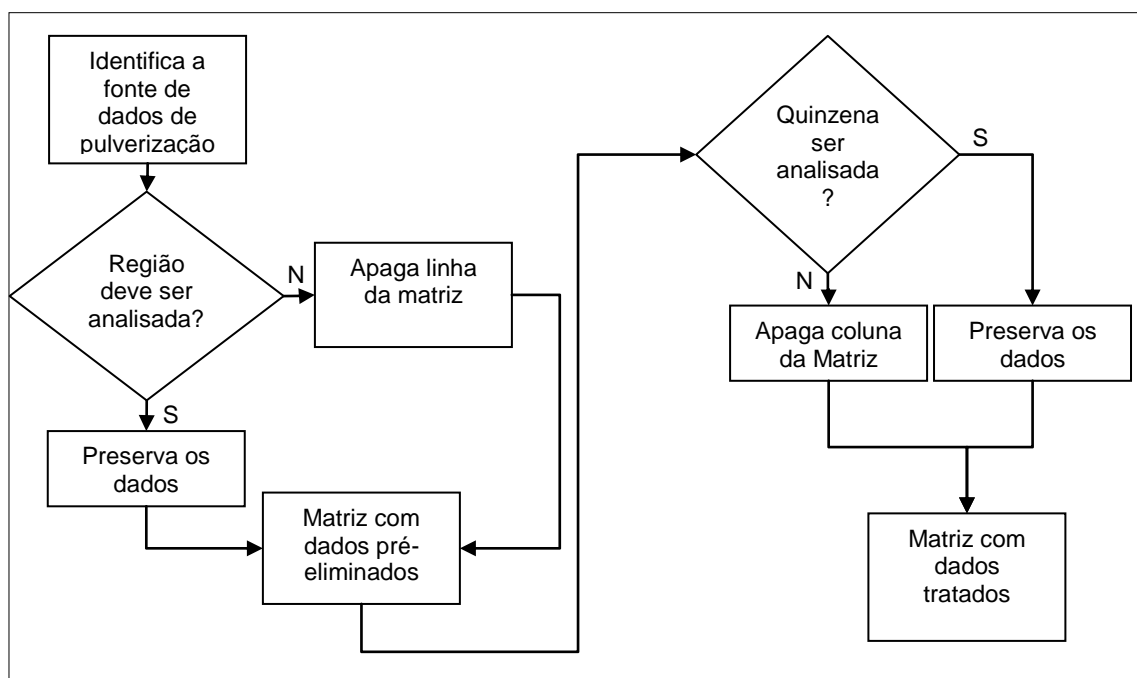


Figura 17 - Procedimento realizado pelo algoritmo de tratamento dos dados brutos de manejo de pragas

4.3. Transformação dos dados em estados

Os estados puderam ser determinados a partir da matriz de dados tratados tanto para as incidências quanto para a pulverização. Enquanto a matriz de dados é quantitativa, a matriz de estados é booleana. Sua construção é realizada em duas etapas: A identificação dos estados de alta e baixa incidência; e a inserção da pulverização para os modelos das abordagens 4.1.2 e 4.1.3.

A identificação dos estados de alta ou baixa incidência é realizada com a comparação dos valores da matriz de dados tratados com os valores críticos definidos na Tabela 5. Caso seja superior ao valor crítico, a posição correspondente recebe valor 1, caso contrário o valor 0 é atribuído à posição correspondente. Ver Figura 18 para a abordagem 4.1.1.

Para as abordagens com a presença da pulverização, a matriz de pulverização é utilizada para atualizar o estado correspondente da matriz de estados. A matriz de estados,

que até então continha os valores 0 ou 1 (correspondente aos estados 1 e 2), é atualizada com a adição do valor 2 para o estado pulverizado (estado 3 para a abordagem 4.1.2 – ver Figura 19) nas posições correspondente à pulverização, independente se o estado for de alta ou baixa incidência.

Para a abordagem 4.1.3 (ver Figura 20) a posição correspondente à pulverização é substituída por dois possíveis valores diferentes:

- O valor 2 (correspondente ao estado 3) caso o valor seja 0 (baixa incidência) e haja pulverização para aquela posição correspondente;
- O valor 3 (correspondente ao estado 4) caso o valor seja 1 (alta incidência) e haja pulverização indicada na posição correspondente.

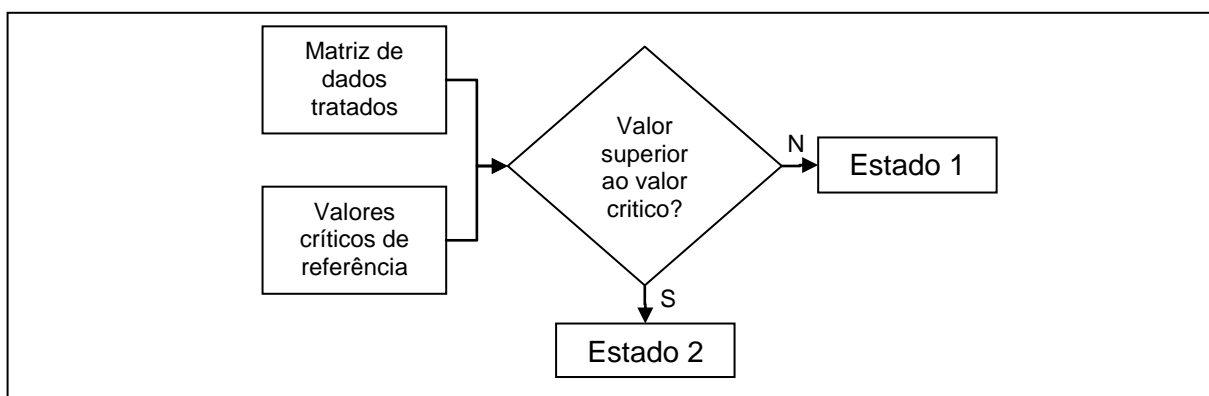


Figura 18 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.1

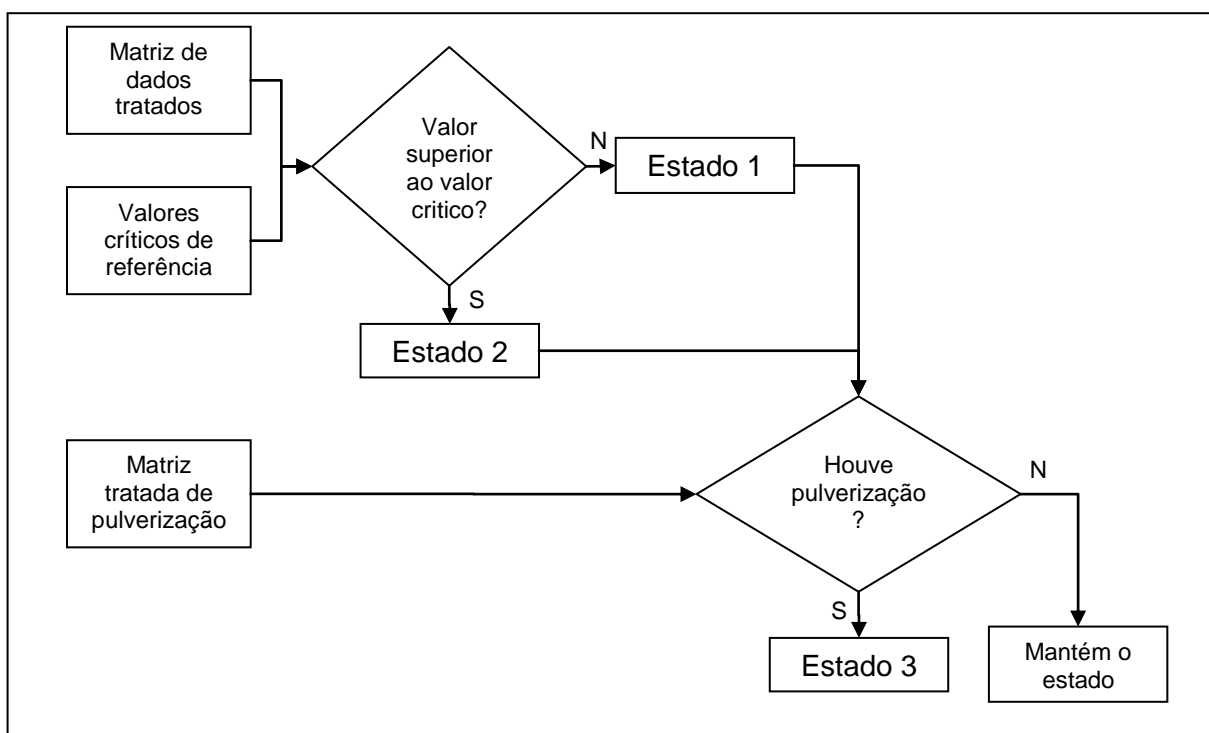


Figura 19 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.2

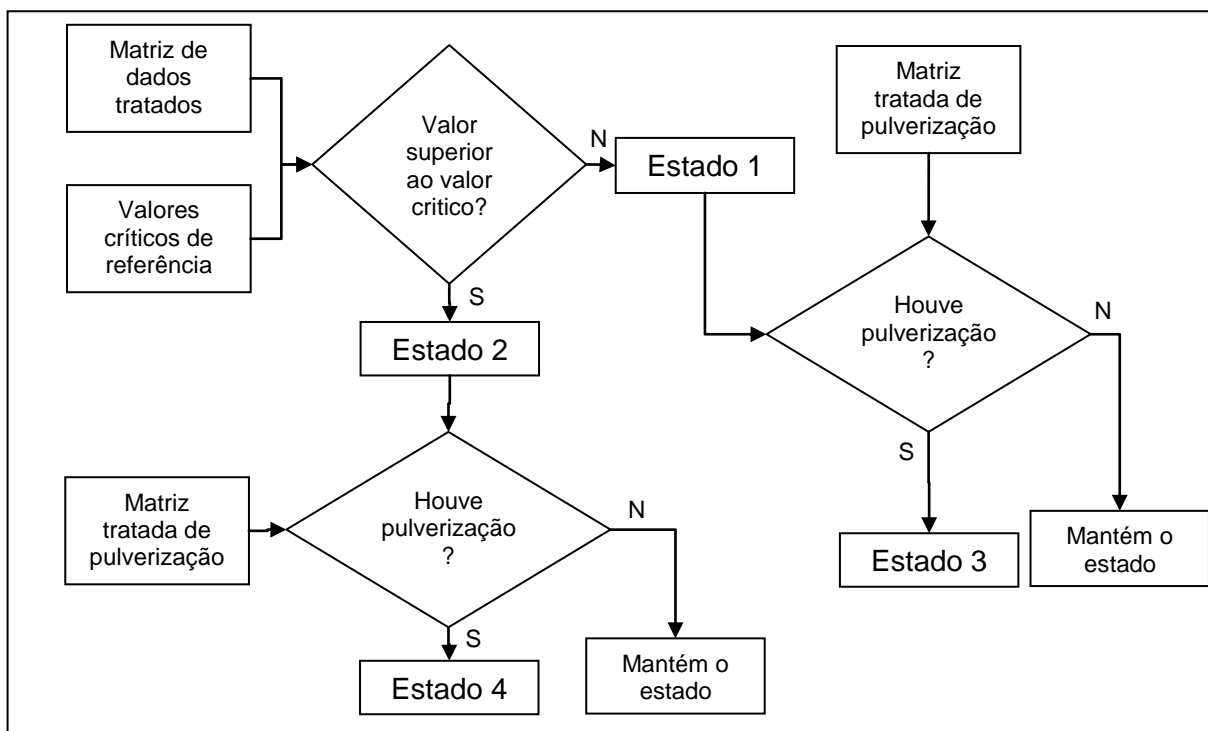


Figura 20 - Algoritmo para a determinação dos estados de acordo com a abordagem 4.1.3

4.4. Determinação das probabilidades de transição em montagem da matriz de transição

Uma vez que os estados foram obtidos, as probabilidades de transição foram calculadas. Para o cálculo destas, é importante lembrar que a sequência de estados foi avaliada, conforme definido na seção 4.1. Por exemplo, se o estado atual, de acordo com o modelo 1, é o estado 1 (baixa incidência) e o próximo estado é o estado 2 (alta incidência), corresponde à transição t_{12} a qual é incrementada. Este procedimento é repetido até que se chegue ao último estado daquela área.

Para quantificação das transições, através de sua frequência relativa, a equação (16), referente à definição de matrizes estocásticas, também foi respeitada e determinante na construção do algoritmo para calcular a matriz de transição. O fluxograma do algoritmo para o modelo de dois estados pode ser visualizado na Figura 21. Este fluxograma é executado para cada uma das 106 áreas definidas e cada uma das pragas relacionadas.

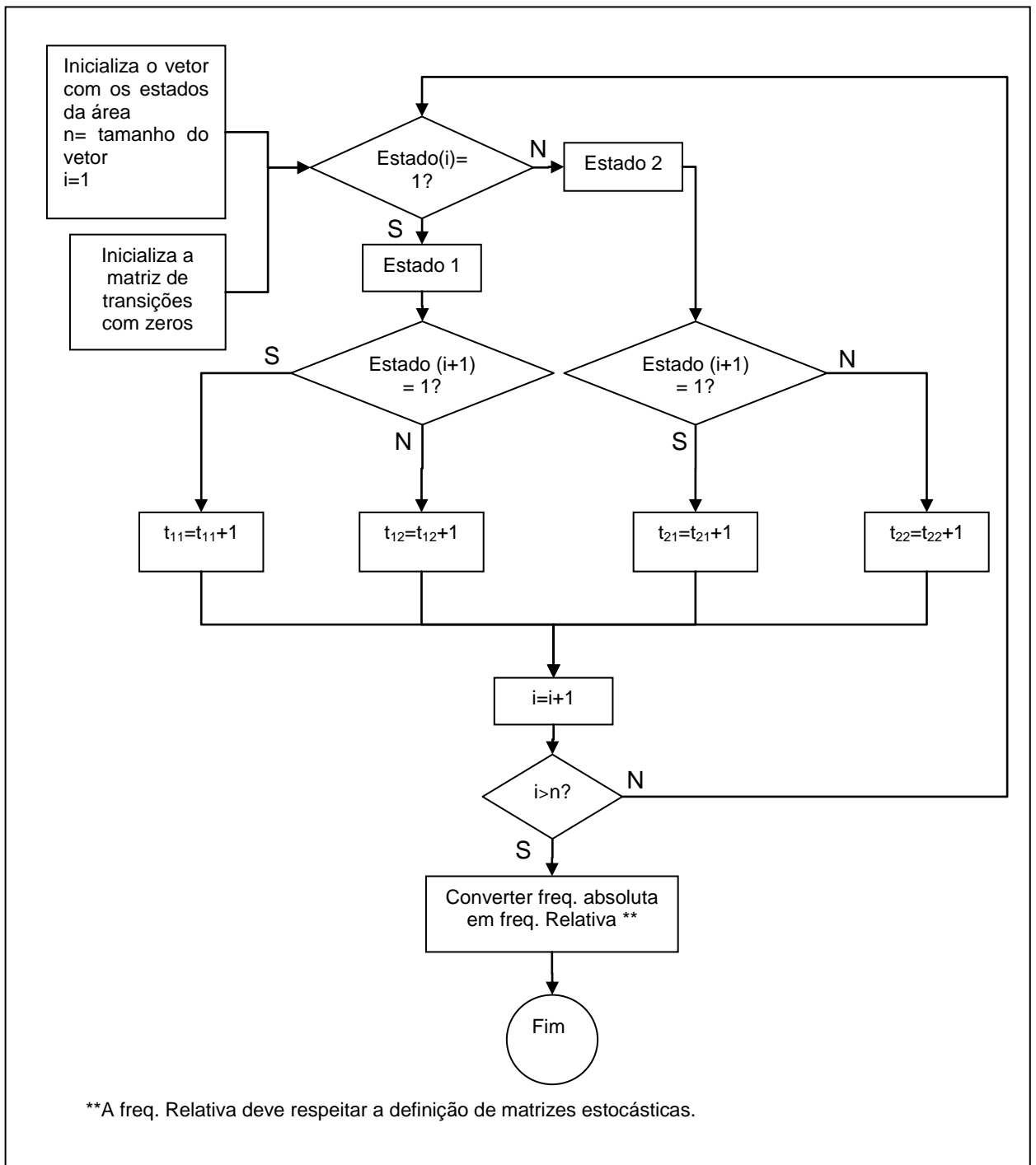


Figura 21 - Algoritmo para determinar a matriz de transição de dois estados (rotina repetida para cada área)

De maneira semelhante foi feito para os modelos de três e quatro estados. A Figura 22 apresenta o fluxograma do algoritmo para a obtenção da matriz de transição para quatro estados. O código do algoritmo de determinação das probabilidades de transição desenvolvido em MatLab® pode ser consultado no Apêndice B. É importante observar que o

período analisado também é variável. Sua variação determina a quantidade de estados (variável n no fluxograma).

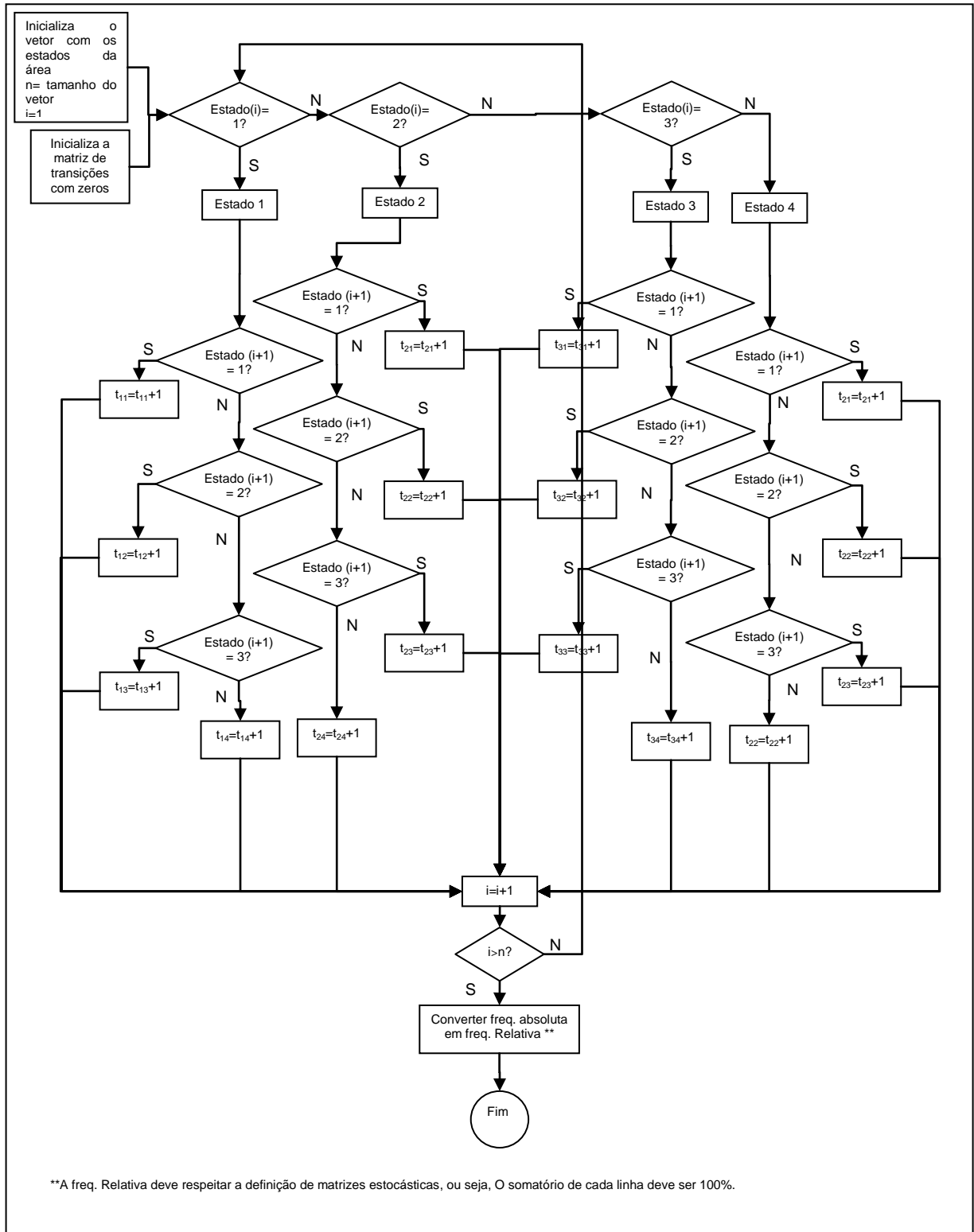


Figura 22 - Algoritmo para determinar a matriz de transição de quatro estados (rotina repetida para cada área)

CAPÍTULO V

RESULTADOS

Após aplicar a metodologia proposta no desenvolvimento do modelo, foram obtidas diversas matrizes de transição. De acordo com o modelo definido, foi encontrada uma matriz de transição referente a cada setor, a cada praga e, a cada ano, totalizando $106 \times 3 \times 3 = 954$ matrizes.

Neste capítulo, serão apresentados alguns resultados pertinentes aos talhões TB26 e TB31 por serem áreas adjacentes para representar e explicar o modelo. Uma comparação entre os modelos também foi realizada utilizando o talhão TB01 setor A. Além disso serão apresentados gráficos de máximos e mínimos por ano de coleta de dados.

O resultado de todas as planilhas está no Apêndice C

5.1. Exemplo de modelo segundo abordagem 1

Para a primeira abordagem, temos apenas dois estados. Um exemplo de construção para este modelo está apresentado a seguir, na Figura 23, em Excel® referente ao talhão TB01, setor A, ano 2011/2012, para a praga1. Esta construção se baseia no algoritmo descrito no capítulo anterior.

A primeira coluna referencia as quinzenas nas quais a amostragem fora realizada e seu valor em frequência relativa na coluna 'praga%'.

A coluna 'estado' se refere à conversão da coluna 'praga%' de acordo com o 'limite%' de baixa incidência indicado para esta praga. Esta coluna serve de base para a montagem das transições, as quatro colunas seguintes.

Estas são então contabilizadas, têm sua frequência absoluta indicada na parte inferior. Em seguida a frequência relativa e a matriz de transição encontrada se localizam na parte inferior. O diagrama de transição para esta solução pode ser visualizado na figura 24.

		LIMITE %	5	1- incidencia baixa 2- incidencia alta				
		SETOR	PRAGA %	ESTADOS	TRANSICOES			
Data					1--1	1--2	2--1	2--2
1	set	A	2	1				
2	set		2,3	1	1			
3	out		3	1	1			
4	out		5,3	2		1		
5	nov		6	2				1
6	nov		7,3	2				1
7	dez		0,3	1			1	
8	jan		0	1	1			
9	jan		0	1	1			
10	fev		0	1	1			
11	fev		0	1	1			
12	mar		0	1	1			
13	mar		0	1	1			
14	abr		1	1	1			
15	abr		1	1	1			
16	mai		1,3	1	1			
17	mai		1,7	1	1			
18	jun		1,3	1	1			
19	jun		0,3	1	1			
20	jul		0,7	1	1			
21	jul		1	1	1			
22	ago		0,7	1	1			
23	ago		0	1	1			
				FREQ ABS	18	1	1	2
						19		3
				FREQ REL	95%	5%	33%	67%
				MATRIZ	SETOR			
					A	95%	5%	
					TB01	33%	67%	

Figura 23 - Modelagem matriz de transição para TB01 - setor A - Praga 1 - ano 2011/2012 – dois estados

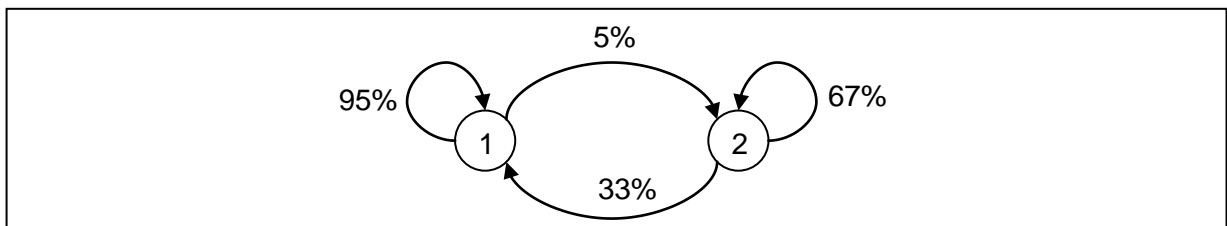


Figura 24 - Diagrama de transição para Praga 1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de dois estados.

5.2 Exemplo de modelo segundo abordagem 2

Introduzido agora o terceiro estado, foi recalculada a matriz de transição para a mesma região. Na Figura 25 encontra-se ilustrado o cálculo da matriz seguindo os passos descritos no capítulo de desenvolvimento do modelo. Observou-se que agora três estados estiveram presentes: 1 – estado de baixa incidência, 2 – estado de alta incidência e 3 – estado pulverizado.

	Data	LIMITE	5	ESTADO	TRANSICOES								
		SETOR	PRAGA		1--1	1--2	1--3	2--1	2--2	2--3	3--1	3--2	3--3
1	set	A	2	1									
2	set		2,3	1	1								
3	out		3	1	1								
4	out		5,3	2		1							
5	nov		6	3						1			
6	nov		7,3	2								1	
7	dez		0,3	1				1					
8	jan		0	1	1								
9	jan		0	3			1						
10	fev		0	1							1		
11	fev		0	1	1								
12	mar		0	3			1						
13	mar		0	1							1		
14	abr		1	1	1								
15	abr		1	1	1								
16	mai		1,3	1	1								
17	mai		1,7	3			1						
18	jun		1,3	1							1		
19	jun		0,3	1	1								
20	jul		0,7	1	1								
21	jul		1	1	1								
22	ago		0,7	1	1								
23	ago		0	1	1								
				FREQ ABS	12	1	3	1	0	1	3	1	0
					55%	5%	14%	5%	0%	5%	14%	5%	0%
					16			2			4		
				FREQ REL	75%	6%	19%	50%	0%	50%	75%	25%	0%
				MATRIZ	SETOR								
					A	75%	6%	19%					
					TB01	50%	0%	50%					
						75%	25%	0%					

Figura 25 - Modelagem matriz de transição para TB01 - setor A - Praga 1 - ano 2011/2012 – três estados

A coluna 'estado', referente à conversão da coluna 'praga%' de acordo com o 'limite%' de baixa incidência indicado para esta praga, apresenta agora os valores 1, 2 e 3. As colunas para a montagem das matrizes de transição são mais numerosas, contabilizando nove, vista a adição de mais um estado. Estas são então contabilizadas têm sua frequência absoluta é também indicada na parte inferior. Em seguida a frequência relativa e a matriz de transição encontrada se localizam, da mesma maneira, na parte inferior. O diagrama de transição referente à matriz encontrada pode ser visualizado na Figura 26.

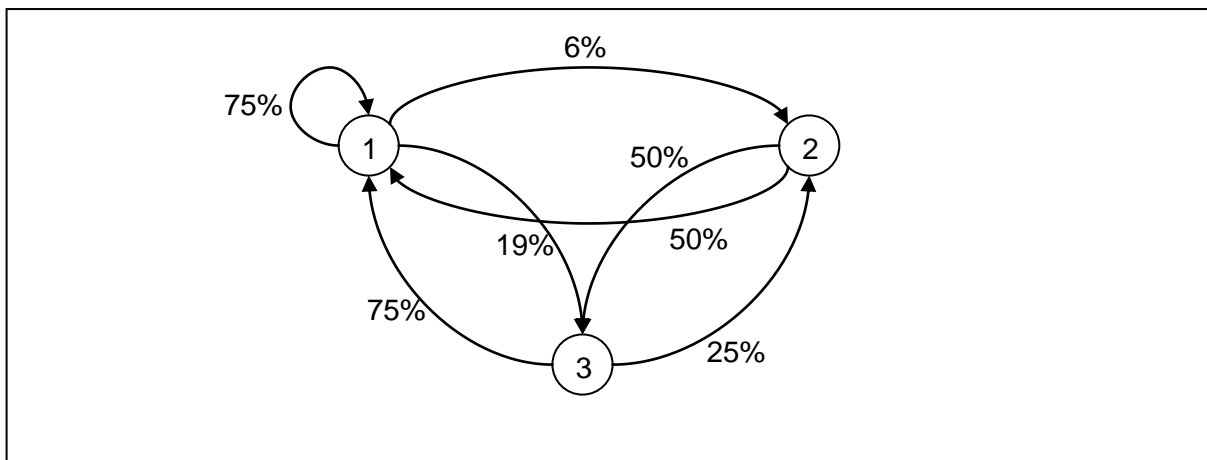


Figura 26 - Diagrama de transição para Praga1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de três estados

5.3 Exemplo de modelo segundo abordagem 3

Finalmente para o modelo de quatro estados, tem-se a pulverização e a incidência sendo analisadas ao mesmo tempo. O talhão TB01 - setor A também foi modelado segundo a abordagem de quatro estados, o que resultou no diagrama ilustrado na Figura 27.

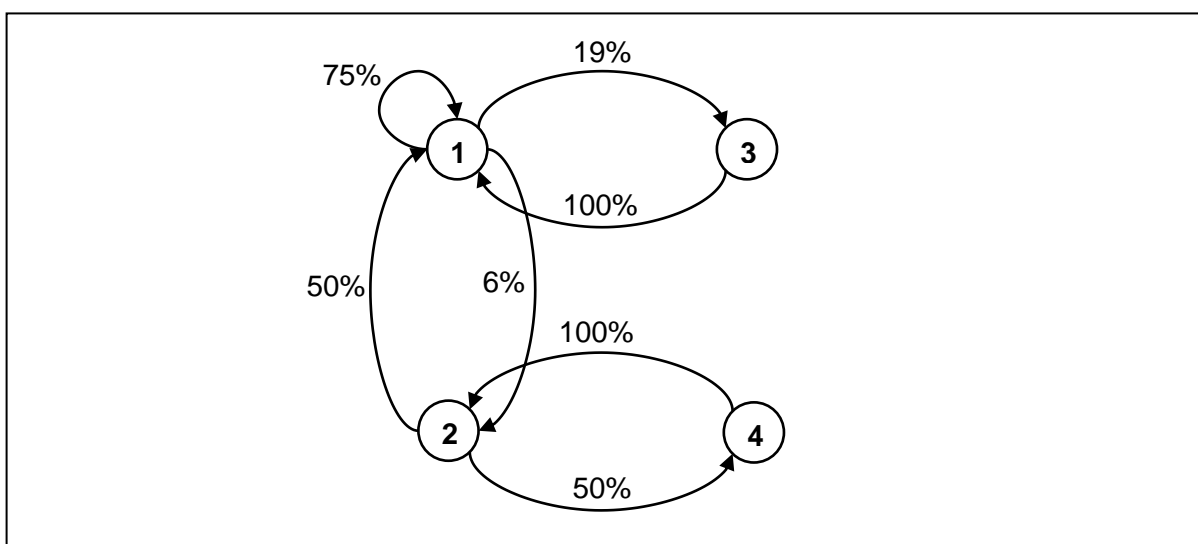


Figura 27 - Diagrama de transição para Praga1, Talhão TB01, setor A, ano 2011/2012 – referente ao modelo de quatro estados

5.4 TB 26 e Tb 31 – Apresentando regiões vizinhas

Foi ainda verificado o comportamento de regiões vizinhas entre si. Para analisar este comportamento foram escolhidas, ao acaso, as regiões TB26 e TB31. Para as quais foram

analisadas as matrizes de transição de cada um dos três anos. A Figura 28 ilustra as regiões analisadas.

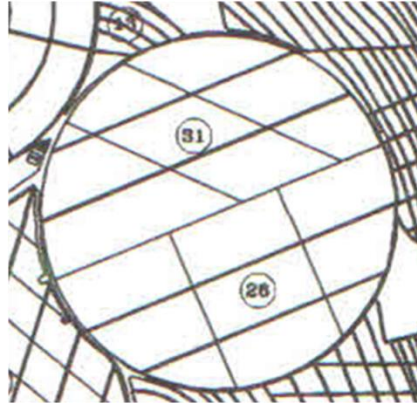


Figura 28 - Regiões vizinhas analisadas

Uma ilustração mais detalhada pode ser visualizada nas Figuras 29 e 30. Nelas podem ser identificados os setores, a quantidade de ruas e de pés de café plantados.

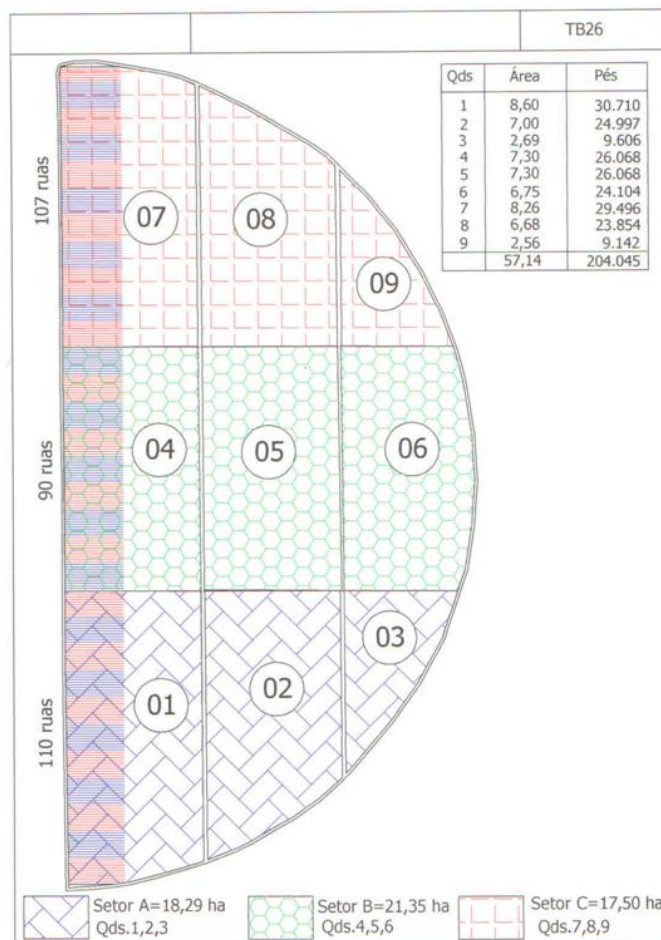


Figura 29 - Ilustração TB26 detalhada

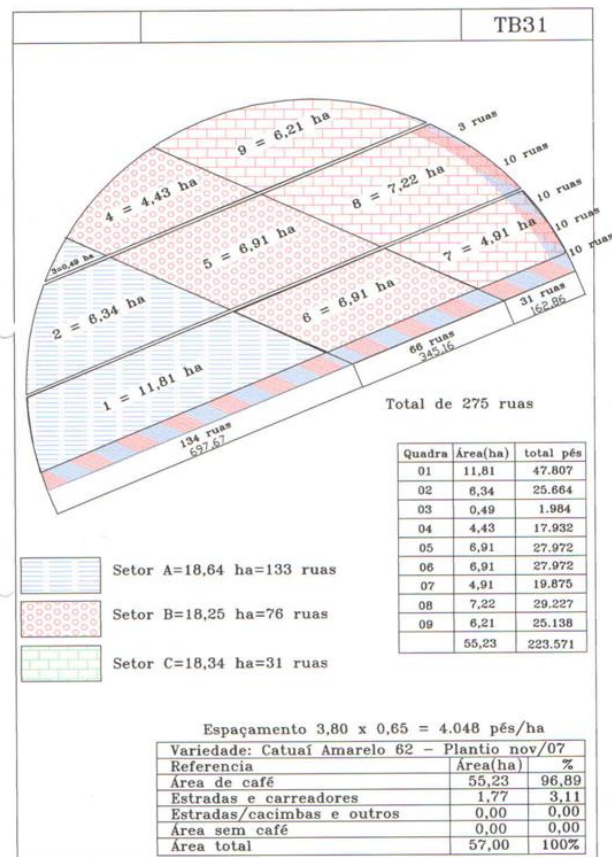


Figura 30 - Ilustração TB31 detalhada

Como pode ser observado, cada uma das duas regiões foi subdividida em 9 quadrantes, os quais foram agrupados de 3 a 3 formando áreas A, B e C de cada um dos talhões. Logo temos três matrizes de transição para cada um dos talhões, para cada ano analisado. O diagrama de transição, referente a cada uma das matrizes, pode ser visualizado nas Figuras 31 - 36. As Figuras 31 e 32 ilustram os diagramas de transição para o ano 2010/2011 para os talhões TB26 e TB31 respectivamente.

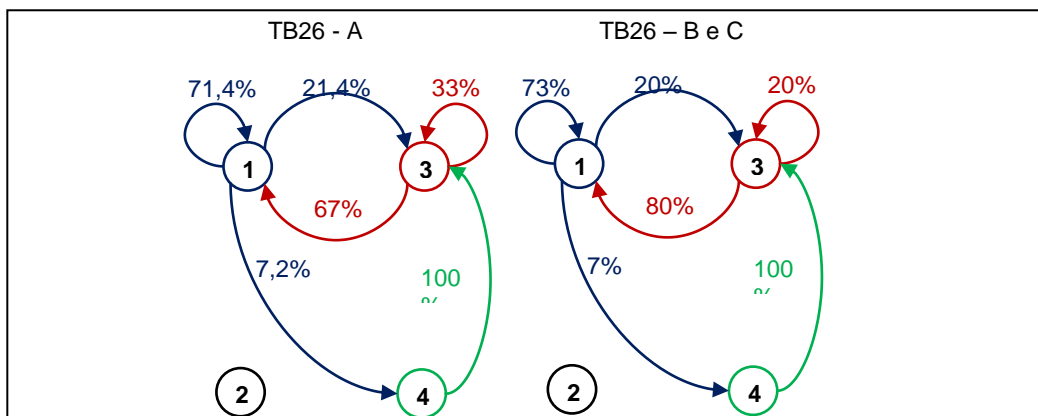


Figura 31 - Diagrama de transição para Praga1, Talhão TB26, ano 2010/2011 – referente ao modelo de quatro estados.

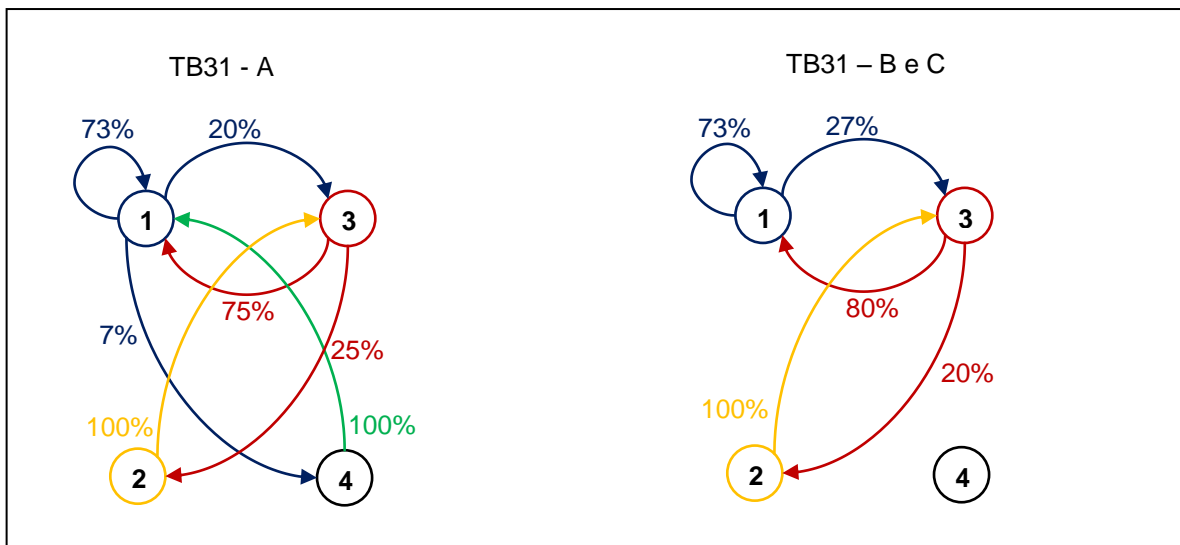


Figura 32 - Diagrama de transição para Praga1, Talhão TB31, ano 2010/2011 – referente ao modelo de quatro estados.

As Figuras 33 e 34 ilustram os diagramas de transição para o ano 2011/2012 para os talhões TB26 e TB31 respectivamente

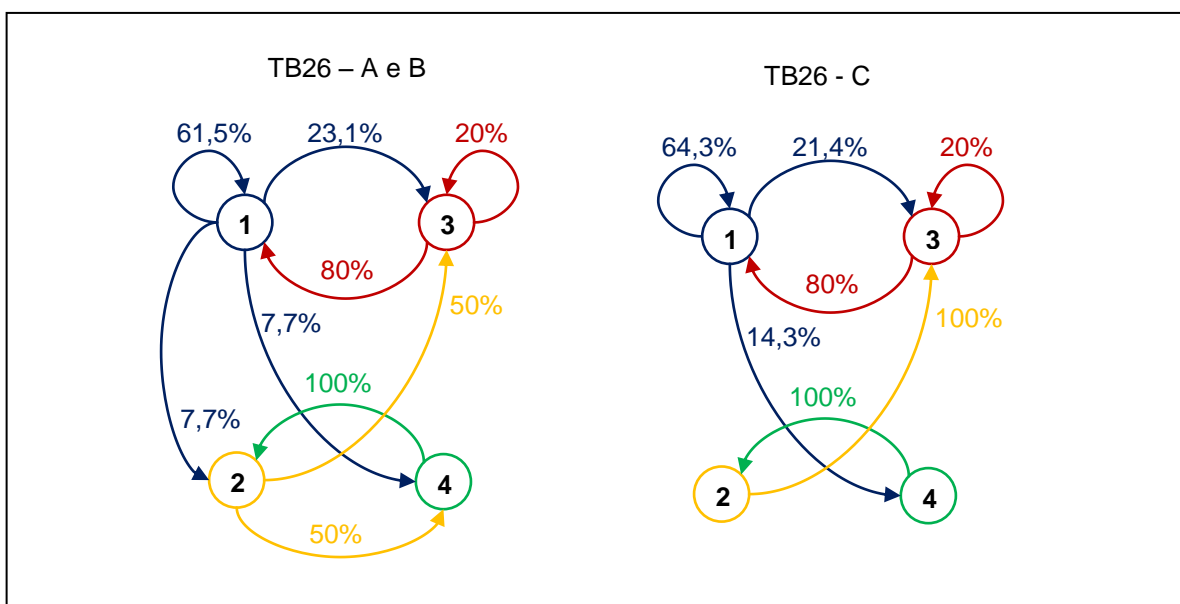


Figura 33 - Diagrama de transição para Praga1, Talhão TB26, ano 2011/2012 – referente ao modelo de quatro estados.

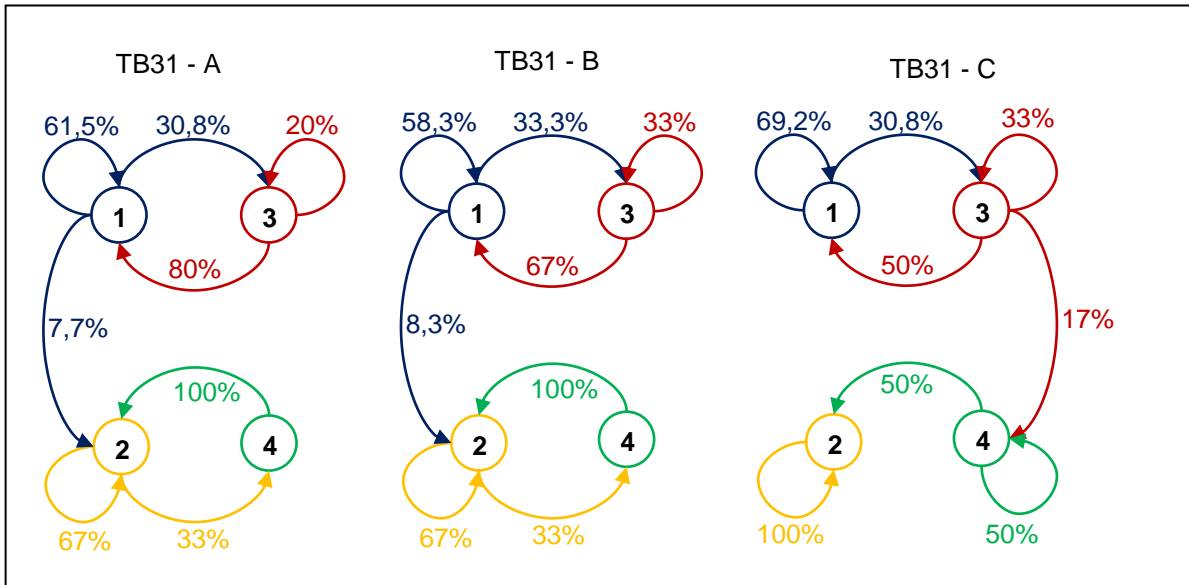


Figura 34 - Diagrama de transição para Praga1, Talhão TB31, ano 2011/2012 – referente ao modelo de quatro estados

As Figuras 35 e 36 ilustram os diagramas de transição para o ano 2012/2013 para os setores TB26 e TB31 respectivamente

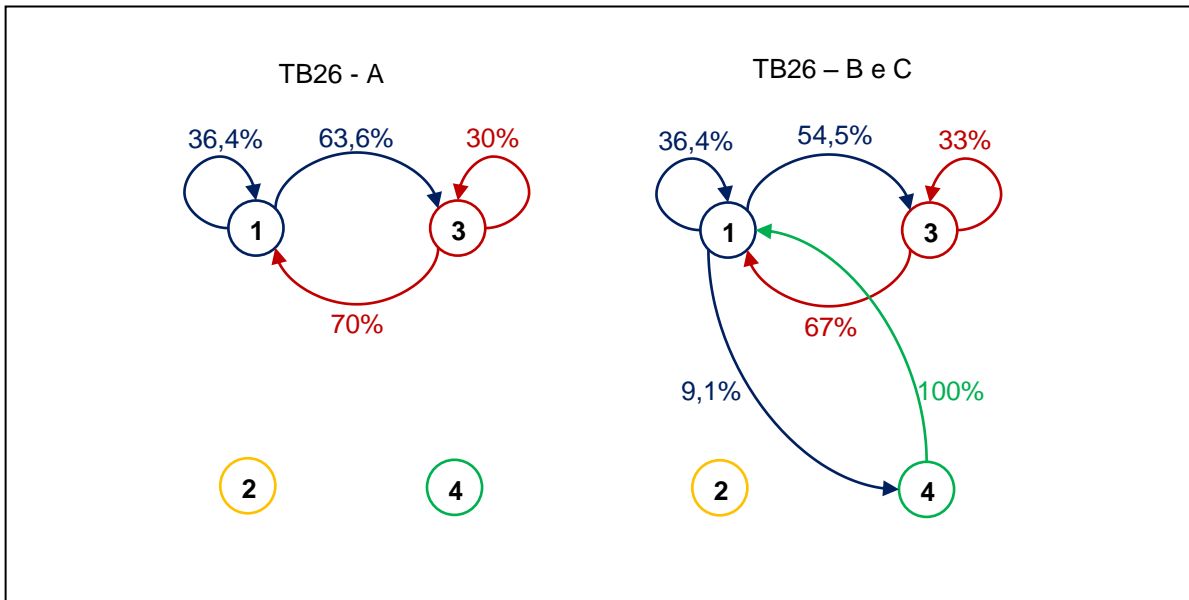


Figura 35 - Diagrama de transição para Praga1, Talhão TB26, ano 2012/2013 – referente ao modelo de quatro estados

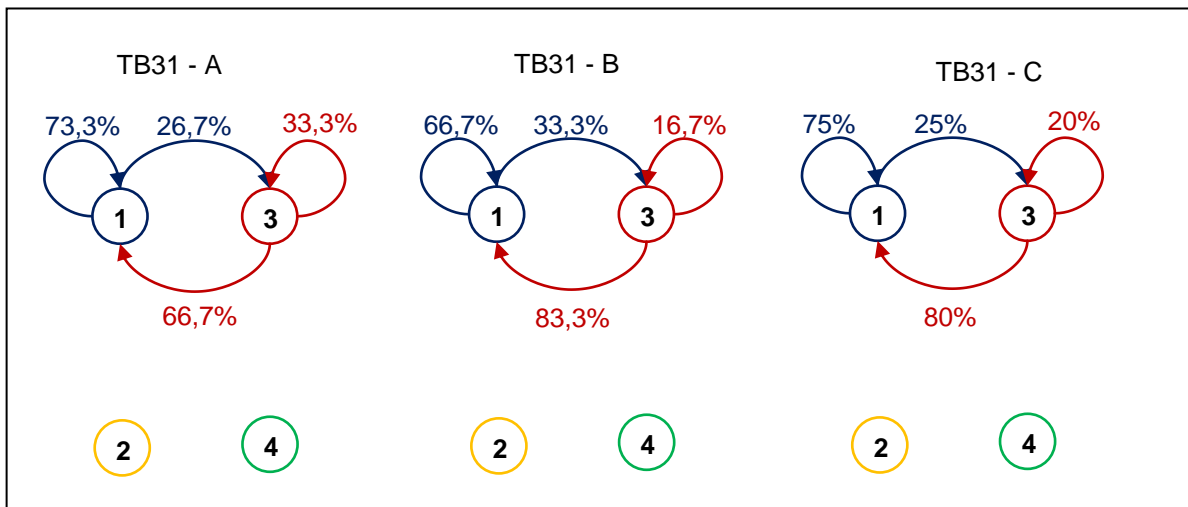


Figura 36 - Diagrama de transição para Praga1, Talhão TB31, ano 2012/2013 – referente ao modelo de quatro estados

Para analisar a permanência no estado de baixa incidência, a transição responsável por este foi colocada no mapa ilustrativo. As Figuras 37, 38 e 39 a seguir mostram os percentuais da transição t_{11} referentes à praga 1 para as regiões vizinhas analisadas.

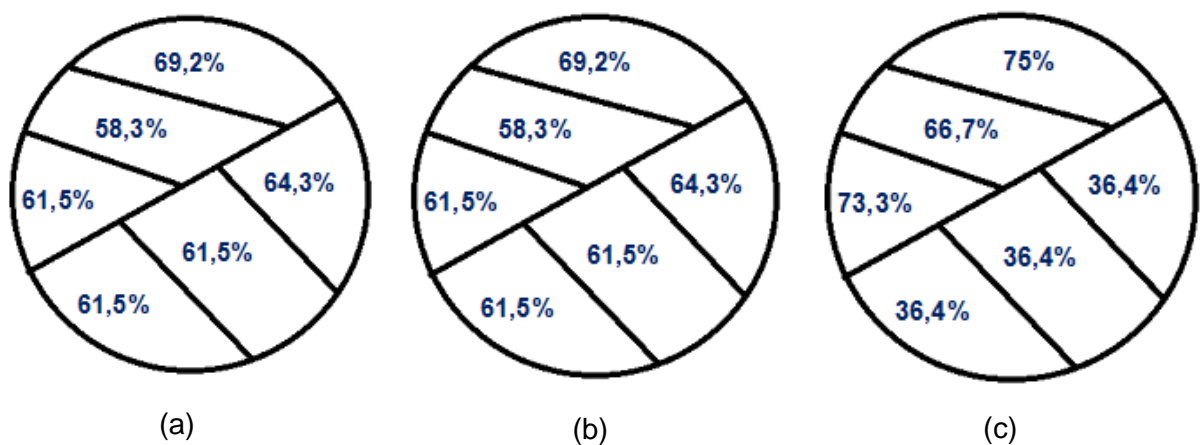


Figura 37 - Índices da transição t_{11} para o ano de (a) 2010/2011 (b) 2011/2012 (c) 2012/2013

Para auxiliar na discussão dos dados apresentados, foi ainda analisado a porcentagem de ocorrência dos estados 1 e 2, representados na Figura 38. Isso para discutir a respeito da não aplicação de insumos.

Foi também analisado o percentual de ocorrência dos estados 1 e 3, para analisar a permanência do estado de baixa incidência nos TBs 26 e 31, conforme representado na Figura 39.

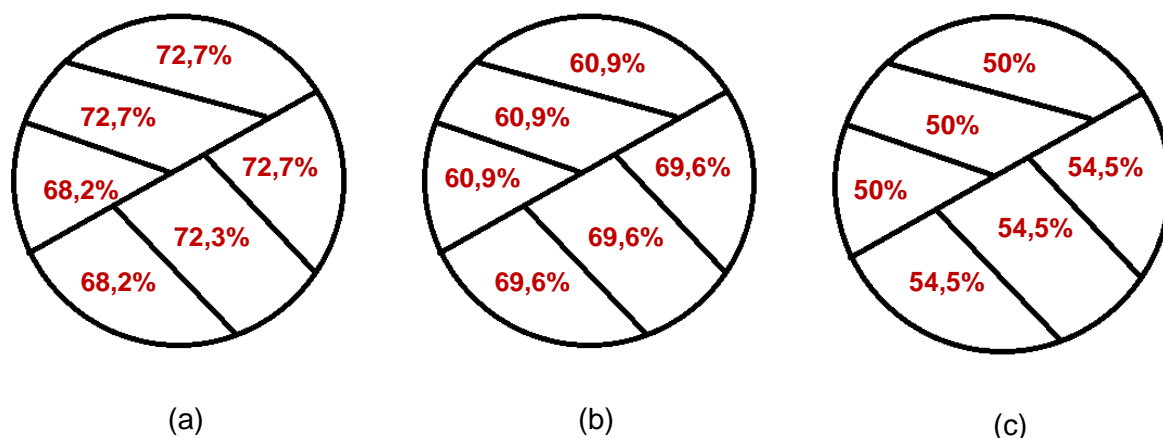


Figura 38 - Percentuais de ocorrência dos estados 1 e 2 para (a) 2010/2011 (b) 2011/2012 (c) 2012/2013

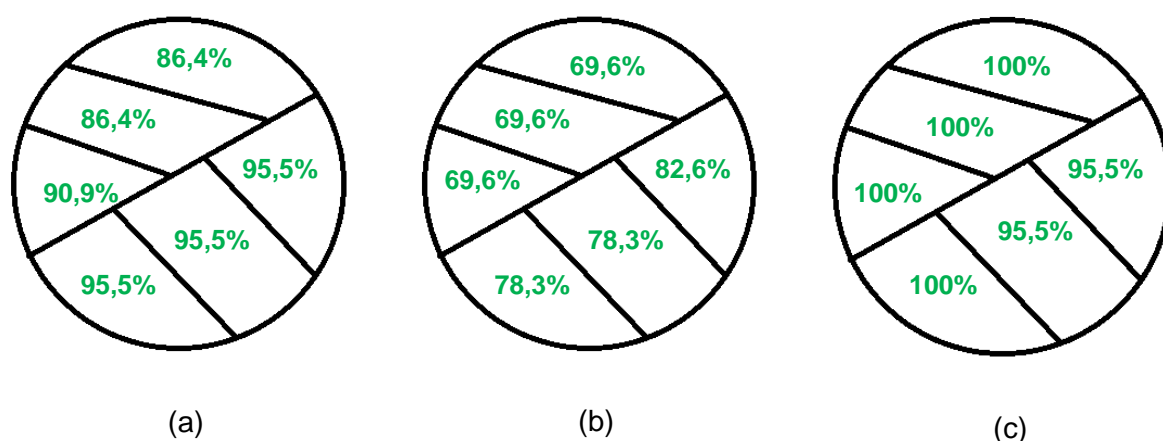


Figura 39 - Percentuais de ocorrência dos estados 1 e 3 para (a) 2010/2011 (b) 2011/2012 (c) 2012/2013

Os gráficos apresentados a seguir referem-se aos talhões TB26 e TB31. As Figuras 40, 41 e 42 ilustram respectivamente, as porcentagens t_{11} para os anos 2010/2011, 2011/2012 e 2012/2013 para todos os 106 setores da fazenda.

Nota-se que em 2010/2011 e em 2011/2012 o setor 44 (TB18) teve maior percentil dessa transição. Para 2012/2013 o setor de maior percentil foi o setor 40. O setor 4 e 5 de 2012/2013 não pertencem mais à fazenda e por isso teve 100% de t_{11} , o que deve ser desconsiderado.

Para os mínimos, percebe-se que em 2010/2011 os setores 54, 57-59 (TB19) tiveram menor percentil. Para 2011/2012 foram os setores 102 e 103 (TB33) e, os setores 72-74 (TB26) referentes ao ano de 2012/2013 que apresentaram os menores percentis.

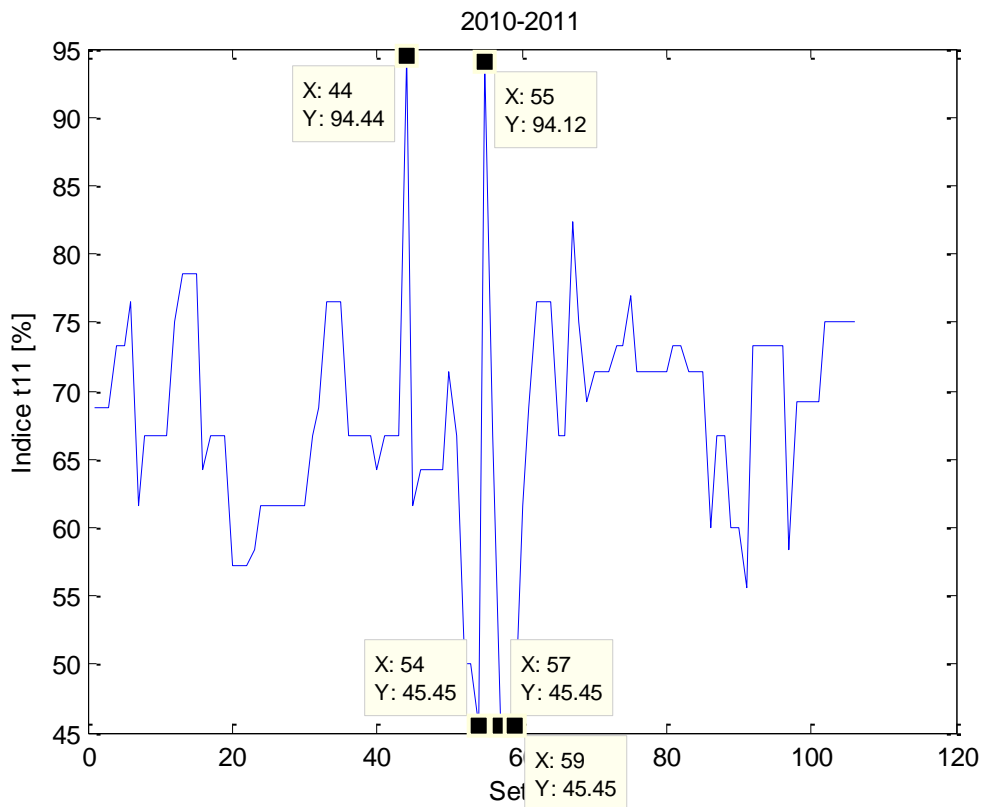


Figura 40 - Porcentagem de transição t_{11} para cada área no ano 2010/2011 referente à praga 1

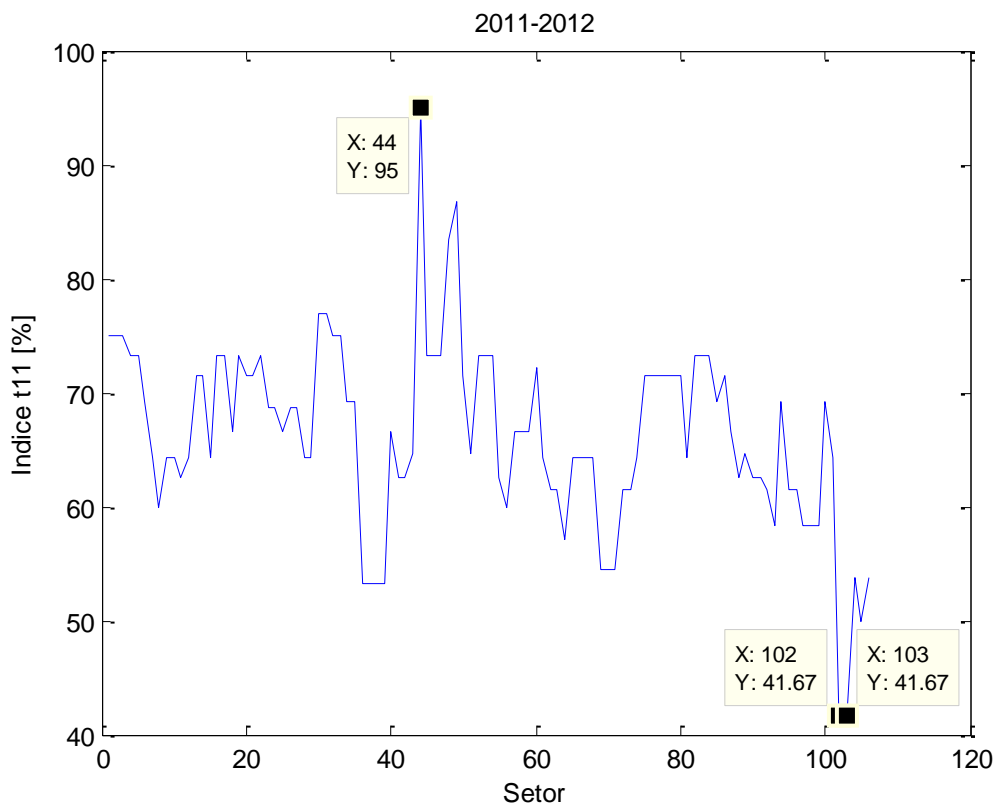


Figura 41 - Porcentagem de transição t_{11} para cada área no ano 2011/2012 referente à praga 1

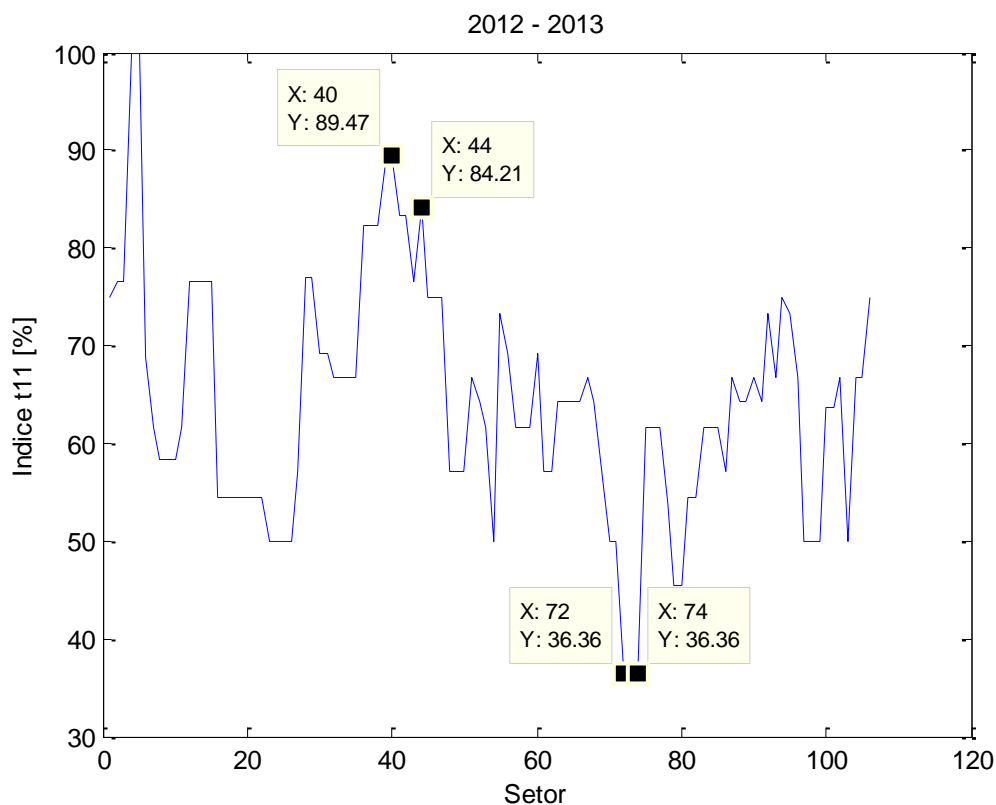


Figura 42 - Porcentagem de transição t_{11} para cada área no ano 2012/2013 referente à praga 1

Também foi verificado em qual dos setores ocorreu o mínimo de incidência e o mínimo de pulverização. Os resultados podem ser visualizados nas Tabelas 7 e 8.

Tabela 7 - Setores com máximo e mínimo de ocorrência dos estados de baixa incidência para o ano 2010/2011

Setores de baixa incidência de pragas - Ocorrência dos estados 1 e 3 - 2010/2011			
	Praga 1	Praga 2	Praga 3
Permanência Mínima	1, 3, 7, 10, 12, 14, 15, 16, 21, 26, 29, 35, 62, 65, 66, 67, 68, 82	12, 14, 15	93, 94, 95, 96, 97
Permanência Máxima	51, 87	102, 103, 104, 105, 106	46

Tabela 8 - setores com máximo e mínimo de ocorrência dos estados sem pulverização para o ano 2010/2011

Setores com baixa pulverização - ocorrência dos estados 1 e 2 - 2010/2011						
	Praga 1		Praga 2		Praga 3	
Pulverização Maxima	44		44		44	
Pulverização Minima	12, 23, 30, 54, 57, 58, 59	12, 23, 30, 54, 57, 58, 59	12, 23, 30, 54, 57, 58, 59	12, 23, 30, 54, 57, 58, 59	12, 23, 30, 54, 57, 58, 59	12, 23, 30, 54, 57, 58, 59

Para finalizar, foi ainda realizada uma comparação ente o modelo geral para toda a fazenda e as médias dos modelos individuais encontrados. Os resultados referentes à praga 1 para os três anos podem ser visualizados na tabela a seguir. Os resultados completos podem ser visualizados no Apêndice D.

Tabela 9: Comparação entre odelo geral e média dos modelos – praga 1

Trans.	2010/2011			2011/2012			2012/2013		
	Modelo Geral	Média dos modelos	Desvio padrão dos modelos	Modelo Geral	Média dos modelos	Desvio padrão dos modelos	Modelo Geral	Média dos modelos	Desvio padrão dos modelos
11	68,69%	67,96%	8,41%	66,82%	66,37%	8,12%	65,97%	64,33%	12,05%
12	0,40%	0,41%	1,73%	1,51%	1,45%	3,27%	0,47%	0,46%	2,19%
13	26,23%	26,90%	9,09%	28,14%	28,56%	8,53%	33,29%	34,89%	11,94%
14	4,67%	4,73%	3,73%	3,53%	3,62%	5,09%	0,27%	0,32%	1,62%
21	40,91%	18,71%	36,76%	43,20%	35,46%	40,65%	62,50%	7,55%	26,54%
22	43,94%	6,98%	20,45%	28,00%	13,44%	24,41%	0,00%	0,00%	0,00%
23	13,64%	5,44%	21,46%	4,80%	2,67%	13,04%	31,25%	4,72%	21,30%
24	1,52%	0,94%	9,71%	24,00%	16,35%	29,18%	6,25%	0,94%	9,71%
31	70,58%	72,58%	22,57%	70,16%	72,04%	16,81%	68,50%	70,05%	21,28%
32	1,40%	1,33%	4,85%	3,32%	3,38%	7,34%	1,13%	1,06%	3,78%
33	27,15%	23,45%	21,24%	23,21%	20,88%	15,71%	29,82%	25,51%	17,47%
34	0,88%	0,76%	3,46%	3,32%	3,70%	8,37%	0,56%	0,94%	2,74%
41	57,14%	43,55%	47,67%	17,92%	12,45%	31,38%	66,67%	5,66%	23,22%
42	12,09%	8,49%	26,25%	64,15%	52,55%	47,27%	22,22%	1,89%	13,67%
43	24,18%	17,14%	35,67%	7,55%	7,55%	26,54%	11,11%	0,94%	9,71%
44	6,59%	2,52%	10,49%	10,38%	4,81%	15,81%	0,00%	0,00%	0,00%

A discussão dos resultados apresentados neste capítulo, pode ser verificada no capítulo a seguir.

CAPÍTULO VI

DISCUSSÃO

6.1 Discussão do modelo

Além dos estados, podem ainda ser observadas as transições. Estas que compõem a matriz de transição de estados e, representam a probabilidade de trânsito entre um estado e outro conforme definido na fundamentação teórica. Para este caso algumas transições podem ser destacadas:

- A transição do estado 1 (baixa incidência e não pulverizado) para o estado 3 (baixa incidência e pulverizado): transição que pode ser interpretada pela aplicação de defensivo agrícola sem a existência da praga, caracterizando uma medida preventiva ou erro de gestão. É uma transição para a qual se espera uma baixa taxa de ocorrência embora seja necessária sua presença a fim de evitar danos na produção;
- A transição do estado 2 (alta incidência e não pulverizado) para o estado 1 (baixa incidência e não pulverizado): transição que representa o sistema naturalmente eliminando a praga incidente sem a intervenção dos defensivos ou com ação tardia da pulverização. É uma transição que caso ocorra, é algo benéfico ao sistema e deve ser detectada e investigada;
- A transição do estado 1 (baixa incidência e não pulverizado) para o estado 2 (alta incidência e não pulverizado): transição que corresponde à infestação da lavoura pela praga, determinando a necessidade do seu controle;
- A transição do estado 2 (alta incidência e não pulverizado) para o estado 4 (alta incidência e pulverizado): transição representando a aplicação do defensivo agrícola na lavoura a fim de controlar a praga;
- A transição do estado 4 (alta incidência e pulverizado) para o estado 1 (baixa incidência e não pulverizado): Esta transição representa a eficácia da aplicação do defensivo agrícola. O ideal é que seu valor seja 100%, indicando que o defensivo é altamente eficaz no controle da praga;

O diagrama de estados deste modelo pode ser visualizado na Figura 43. Algumas transições foram marcadas em vermelho pontilhado, pois são transições que não se deseja observar no sistema. Por exemplo, a transição do estado 4 para o estado 3, o estado de baixa incidência foi atingido porém o defensivo foi ainda ministrado na lavoura, representando desperdício. A transição t_{33} também representa que aplicações sequenciais de defensivos estão sendo aplicadas sem a necessidade das mesmas. Já as transições t_{44} e t_{42} indicam a ineficácia do insumo aplicado ou o aumento da resistência da praga junto ao defensivo.

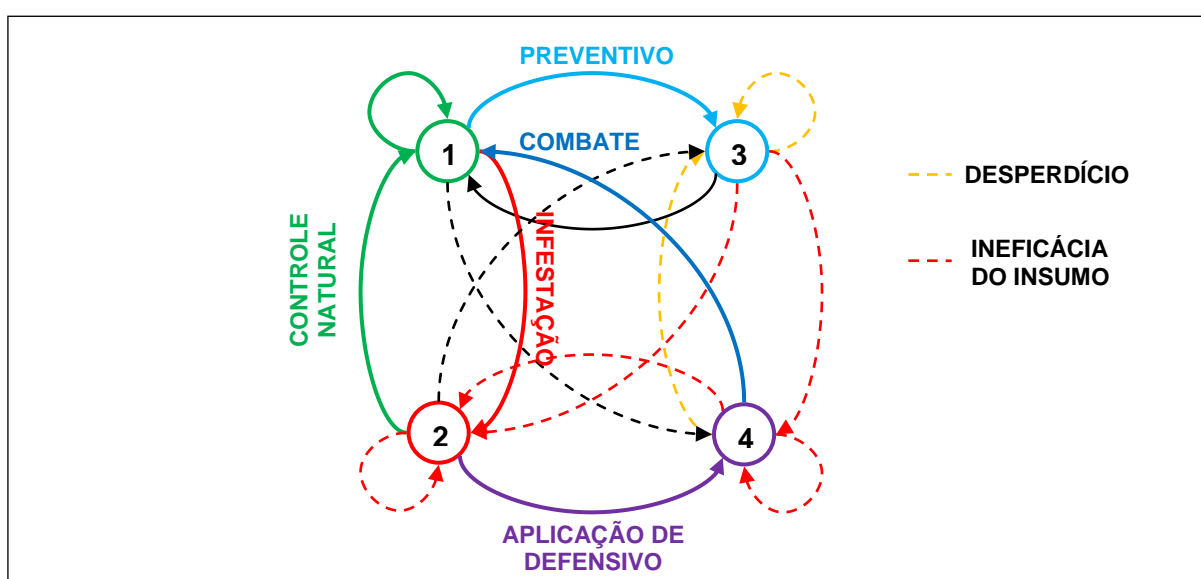


Figura 43 - Diagrama de estados da segunda solução proposta com interpretação das transições.

6.1. Discussão dos resultados

6.1.1 Comparação entre os modelos

Para o modelo de dois estados pôde ser observado que há um período grande de ocorrência do estado 1 (estado de baixa incidência da praga) e apenas alguns estados 2 (estado de alta incidência da praga). Observando a matriz de transição, percebeu-se um grande percentual da transição t_{11} conforme esperado e, baixo valor de t_{12} , representando uma ação da praga sobre a área A em pequena frequência. Porém observa-se um problema com a transição t_{21} , indicando que o sistema permaneceu em estado de alta incidência, o que é prejudicial à lavoura.

Este modelo informa a respeito da transição entre estados de alta ou baixa incidência mas, não permite dizer nada a respeito das pulverizações realizadas. Mesmo que se infira, por ter ocorrido um alto percentual da transição t_{11} que tenham ocorridas poucas

pulverizações, um conclusão errada pode ser tirada deste cenário. Uma tentativa de tentar evitar a possibilidade desse erro foi introduzir o terceiro estado: o estado pulverizado.

Pelo diagrama do terceiro estado apresentado, para a mesma região referente ao mesmo período, puderam ser percebidos alguns problemas para esta área no período indicado, o que não estava evidente no modelo de dois estados:

- A presença da transição $t_{32} = 25\%$, indicando um retorno ao estado infestado após a pulverização. Pode ser interpretado como ineficácia do insumo aplicado;
- Pulverizações realizadas sem necessidade estão indicadas através da transição $t_{13} = 19\%$. Para esta transição há duas possíveis interpretações: Uma aplicação preventiva ou um erro de aplicação. É difícil concluir a respeito, uma vez que o estado de alta ou baixa incidência não está vinculado ao estado pulverizado. Na prática o estado pulverizado acontece em paralelo à incidência, motivo o qual levou à terceira abordagem com quatro estados.

Aplicado o modelo de quatro estados, puderam ser diferenciadas as pulverizações corretivas das preventivas ou excessivas. Como temos os estados 3 - baixa incidência e pulverizado e 4 - alta incidência e pulverizado, as medidas corretivas puderam ser observadas pelas transições t_{13} e as preventivas através de t_{24} . Para t_{13} foi observado um percentual considerável, o que pode ser interpretado como aplicações além do necessário.

Uma vez atingido o estado de alta incidência (estado 2), somente dois caminhos foram tomados com igual percentual (50%): retornar à baixa incidência (estado 1) ou aplicar uma pulverização corretiva (estado 4).

Uma transição preocupante é a $t_{42} = 100\%$, indicando que o sistema retornou ao estado de alta incidência após a aplicação do insumo. Isto indica uma ineficácia do insumo aplicado. Confirmando esta ocorrência com a fonte dos dados, foi dito que esse fato realmente foi encontrado na fazenda.

6.1.2 Análise das regiões vizinhas

Para a análise das regiões vizinhas, foram escolhidas duas regiões adjacentes: o TB26 e o TB31. Conforme pode ser visto nos diagramas de transição os comportamentos dentro do mesmo talhão, para a mesma época, se mostraram semelhantes. Apenas pequenas diferenças no valor percentual podem ser verificadas ao compararmos os diagramas dentro de cada ano.

Comparando os períodos 2010/2011, 2011/2012 e 2012/2013 para uma mesma região, percebeu-se uma variação nos diagramas de transição. Notou-se uma diminuição do valor da transição t_{11} a qual pode ser atribuída a duas razões diferentes a cada ano:

- Para 2011/2012 notou-se a ocorrência do estado de alta incidência, que se justifica pela Figura 39(b);
- Para 2012/2013 notou-se que houve aplicação de insumo, embora não se tenha ou se tenha em pequeno percentual a ocorrência do estado de alta incidência, justificado pela Figura 39(c).

Percebeu-se ainda na Figura 34, a ocorrência de um estado absorvente (2). A presença deste estado pode ser justificada pela ocorrência do estado 2 no final do período. A presença da transição t_{22} indica danos à lavoura, comprometendo a produtividade e o preço dos produtos.

Pode-se perceber ainda, pela análise das Figuras 38 e 39, que a maior incidência de pragas foi observada em 2010/2011 e 2011/2012 no talhão TB31. Para o ano de 2012/2013 foi encontrado 100% de baixa incidência, o que acorda com o decréscimo de percentis sem pulverização (Figura 38) e indica a eficiência do insumo aplicado. Também pode ser observado que houve um decréscimo da alta incidência em virtude do aumento da pulverização, quando comparadas as Figuras 38 e 39.

Analisando agora globalmente as transições t_{11} , percebe-se que os máximos encontrados para dois anos seguidos referem-se ao TB18 (setor 44). Coincidentemente, a tabela 8 mostra que este talhão apresenta o máximo de estados sem pulverização para os três anos analisados. Como este talhão realiza a produção de café orgânico, todos os fatos observados e apresentados se justificam e credibilizam a validade da ferramenta apresentada.

Para os mínimos encontrados, estes não apresentam um padrão, uma vez que se observam diferentes áreas apresentando o mínimo em cada ano. O TB19 (setores 54, 57-59) é uma região muito extensa e com diferentes tipos de café e com diferentes idades de plantio. Logo é justificável que haja variação nas regiões vizinhas e outros dados são necessários para melhor análise.

Pode ser observado, pelas tabelas 7 e 8, que setores com alta pulverização obtiveram mínima incidência, como o setor 12, por exemplo. Tal fato observado evidencia que fatores controláveis como a pulverização influenciam em resultados evidentes e esperados. O mesmo não pode ser dito em relação à incidência, que possui fatores que ainda não identificamos, fato que nos impede de controlar ou prever a sua tendência.

6.1.3 Explosão de estados

Visto que são diversas áreas consideradas e diferentes pragas, diversos cenários podem ser construídos. Vamos analisar para o modelo mais simples, o modelo da abordagem 4.1.1. Sendo 114 setores analisados, para apenas 1 praga por cada área e, analisando apenas a presença/ausência da praga, pode-se utilizar a análise combinatória para calcular a explosão de estados possíveis. Pela linha 114 do triângulo de Pascal, ver Figura 41, que corresponde a todas as combinações possíveis de incidência desta praga pelos 114 setores, pode ser realizado o seguinte cálculo:

$$C_{114,0} + C_{114,1} + \dots + C_{114,114} = 2^{114} \cong 2,07 \times 10^{34} \text{ (dimensão do vetor de estados)} \quad (29)$$

$$\begin{array}{c}
 \binom{0}{0} \\
 \binom{1}{0} \quad \binom{1}{1} \\
 \binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2} \\
 \binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3} \\
 \binom{4}{0} \quad \binom{4}{1} \quad \binom{4}{2} \quad \binom{4}{3} \quad \binom{4}{4} \\
 \dots \\
 \binom{n}{0} \quad \binom{n}{1} \quad \binom{n}{2} \quad \binom{n}{3} \quad \binom{n}{4} \dots \binom{n}{n} \\
 \dots
 \end{array}$$

Figura 44 - Triângulo de pascal

Fonte: <http://www.somatematica.com.br/emedio/binomio/binomio2.php> Acessado em: 18/12/2014 as 22h10min

Isto corresponde a um número consideravelmente grande de estados possíveis, ainda que fora considerada apenas uma praga para o modelo mais simples (dois estados). A matriz de transição correspondente a este número de estados teria aproximadamente $4,31 \times 10^{68}$ transições. O esforço computacional para determinação desta matriz é enorme e inviável, dados os recursos disponíveis e a praticidade do modelo.

CAPÍTULO VII

CONCLUSÃO

O presente trabalho teve como objetivo a modelagem de um sistema com o uso da cadeia de Markov. Após todo o estudo e definições feitas, chegou-se a uma definição, a partir da qual foi proposta uma metodologia e três abordagens diferentes foram determinadas. Para os modelos apresentados, observou-se uma evolução quanto aos estados analisados, começando com a análise apenas da incidência e terminando com a análise das pulverizações realizadas em paralelo com a incidência observada.

Observando os modelos, vemos que a análise começou em uma dimensão e terminou em duas. Essas duas dimensões foram definidas segundo os dados que estavam disponíveis para análise, a fim de modelar o problema proposto.

Claro que para cada informação analisada, acrescenta-se uma dimensão a mais ao modelo, aumentando o número de estados possíveis. Assim tem-se uma matriz de transição cada vez mais complexa. Conforme apresentado na discussão dos dados, a própria definição do modelo de maneira segmentada por praga e por área, foi feita a fim de tornar a matriz menos complexa.

A metodologia desenvolvida e implementada neste trabalho mostrou-se bastante efetiva, conduzindo a um resultado satisfatório no que diz respeito à obtenção da ferramenta para auxílio no monitoramento de pragas. O resultado encontrado é claramente um sistema de apoio a decisão e monitoramento de praga que permite a identificação de falhas no manejo, desperdício, efetividade na aplicação dos insumos, dentre outras possibilidades de análise.

Observou-se ainda que as matrizes tiveram variação ao longo dos anos. Realizada as comparações entre o modelo geral e a média dos modelos individuais, pode-se dizer que todos estão dentro do comportamento esperado do sistema. Logo podemos considerá-lo como homogêneo.

Para investigar ainda mais a ocorrência da incidência, foi analisada a ocorrência dos estados através de sua frequência relativa. Analisando os valores, percebeu-se que não há um padrão, logo comportamento dos estados realmente é diferente, justificando a aplicação da cadeia de Markov para análise do sistema.

É importante lembrar que apenas duas dimensões foram consideradas. Em se adicionando mais dimensões, pode-se evoluir para um sistema a eventos discretos, que leva a outra abordagem que não a cadeia de Markov.

Pode se afirmar que o modelo encontrado pode ser utilizado como uma ferramenta de supervisão e monitoramento. Os resultados apresentados evidenciaram os problemas de pulverização encontrados na lavoura. Para gestão do sistema e previsão do que se fazer em relação à lavoura, se configura como uma ferramenta interessante. Ainda mais se consideramos um profissional que embora da área, não tenha muita experiência no manejo de pragas. Com o uso da ferramenta, ele pode atestar a eficácia dos insumos utilizados, verificar as aplicações mal sucedidas, a fim de conduzir a um produto final de qualidade e com boa produtividade.

Pode ser observado ainda, o efeito da pulverização na redução das incidências e a possibilidade de homogeneização, sendo este fator controlável. Paralelamente a esta situação temos a não homogeneização das incidências, que possuem comportamento sem um padrão definido por nós, com vários fatores de influência. Além dos fatores já citados, que acrescentam dimensões ao modelo, ainda podem ser colocadas na lista o tipo de café e a idade de plantio. Estes fatores são muito importantes para determinar a resistência do café a certos tipos de pragas. Logo o resultado apresenta cadeias não homogêneas.

A cadeia de Markov é apenas um componente do processo de decisão Markoviano. Para ser montado esse processo de decisão, devem ainda ser analisados os processos de decisão, associados a uma política estacionária. Como a cadeia evolui no tempo, mediante um política estacionaria definida, assim que identificada a cadeia, associada a política estacionaria, a decisão poderia ser tomada.

Assim damos um passo a mais para a agricultura de precisão. A questão é se atentar aos parâmetros que são necessários, mas que não tornem o sistema tão complexo, ao ponto de inviabilizar o uso da ferramenta. Como realizar as previsões, montar o problema de decisão de maneira a prevenir o estado de alta incidência e controlar o nível das pragas com o uso mínimo de defensivos, se possível erradicação.

CAPÍTULO VIII

TRABALHOS FUTUROS

A proposta inicial do desenvolvimento deste trabalho era de desenvolver um software para monitoramento, previsão e controle do nível das pragas. Para se construir o problema de decisão Markoviano, foi necessária a modelagem da cadeia de Markov do sistema, a fim de se definir os estados, conhecer o comportamento para analisar as previsões e definir as decisões.

Com essa ferramenta em mãos, a mesma pode ser introduzida em um software e estudar a viabilidade de embarcá-la em um sistema autônomo, que pode realizar a coleta monitoramento e as pulverizações automaticamente, baseando-se em um problema de decisão que é alimentado pelos próprios dados do sistema, evoluindo com o tempo.

Mas ainda se tratando do próprio desenvolvimento do modelo, o mesmo pode se aperfeiçoado, vista a diversidade de características que existem para ser analisadas no sistema e não o foram, devido a falta de disponibilidade de dados referente às mesmas.

Para uma cadeia de Markov mais completa, deveriam ser acrescentadas as dimensões:

- Umidade
- Altitude
- Incidência solar
- Característica do solo
- Temperatura
- Vento

Ainda para análise financeira poderiam ser acrescentadas as informações:

- Custo de aplicação dos insumos
- Qualidade do café /Preço de venda
- Quantidade produzida

Abordagens alternativas na construção das matrizes de transição podem também ser estudadas, como no trabalho proposto por (Janssens et al., 2005). Esse trabalho elimina possíveis vícios na cadeia, como a repetição de um estado no começo da linha temporal que não acontece depois, comprometendo as probabilidades de transição.

Características como custo médio por unidade de tempo e tempo de recorrência de um estado também podem ser estudados a fim de estudar e auxiliar no problema de decisão a ser proposto.

Outras abordagens envolvem fazer análise de cadeias de Markov de 2 e 3 graus, olhando a influência dos estados $t - 1$ (quinzena anterior), $t - 2$ (mês anterior), $t - 3$ (um mês e meio anterior).

Deve ainda ser considerado o uso da lógica fuzzy para a visualização da incidência. Sendo uma ferramenta para a operação do sistema, é importante buscar a possibilidade do uso da ferramenta para a previsão de incidência. Um grande desafio é realizar planejamentos futuros a partir do modelo obtido.

REFERÊNCIAS BIBLIOGRÁFICAS

APARANA U, Lene Juul PEDERSEN, Erik JORGENSEN. (2014). Hidden phase-type Markov model for the prediction of onset of farrowing for loose-housed sows. In: Computers and Electronics in Agriculture 108 (2014) 135-147.

Disponível em: <<http://dx.doi.org/10.1016/j.compag.2014.07.008>> Acesso em:28/10/2014.

ARSANJANI Jamal Jokar , Marco HELBICH, Wolfgang KAINZ, Ali Darvishi BOLOORANI. (2014). Integration of logistic regression, Markov chain and cellular automata models to simulate urban expansion. In: International Journal of Applied Earth Observation and Geoinformation 21 (2013) 265-275.

Disponível em: <[10.1016/j.jag.2011.12.014](http://dx.doi.org/10.1016/j.jag.2011.12.014)> Acesso em:28/10/2014.

ATACHADÉ Yves F., Matias D. CATTANEO. (2014). A martingale decomposition for quadratic forms of Markov chains (with applications). In: Stochastic Processes and their Applications 124 (2014) 646-677. Elsevier.

Disponível em: <<http://dx.doi.org/10.1016/j.spa.2013.09.001>> Acesso em:28/10/2014.

BARRIGA Gladys D.C, Francisco LOUZADA. (2014). The zero-inflated Conway-Maxwell-Poisson distribution: Bayesian inference, regression modeling and influence diagnostic. In: Statistical Methodology 21 (2014) 23-34.

Disponível em: <<http://dx.doi.org/10.1016/j.stamet.2013.11.003>> Acesso em:28/10/2014.

BEARZOTI Eduardo, Luiz Henrique de AQUINO. (1994). Plano amostragem sequencial para avaliação de infestação de bicho-mineiro (*lepidoptera: Lyonetiidae*) no sul de Minas Gerais. In: Pesquisa agropecuária Brasileira, v.29 n.5 695-705, maio 1994,

Disponível em: <<http://seer.sct.embrapa.br/index.php/pab/article/view/4105>> Acesso em:20/12/2012.

CHENG Qin-Bo , Xi CHEN , Chong-Yu XU , Christian REINHARDT-IMJELA , Achim SCHULTE (2014). Improvement and comparison of likelihood functions for moedl calibration and parameter uncertainty analysis within a Markov chain Monte Carlo schema. In: Journal of Hydrology 519 (2014) 2202-2214.

Disponível em: <<http://dx.doi.org/10.1016/j.jhydrol.2014.10.008>> Acesso em: 28/10/2014.

COUTINHO Darley, Marcelo C. PICANÇO , Ézio M. SILVA ,Ivênio R.OLIVEIRA , Elisangela G. FIDELIS , Carla C. MILAGRES (2005). Amostragem sequencial de *leucoptera coffeella* em cafeeiros em produção pela contagem de minas com lagartas em folhas do terço mediano. In: Simpósio de Pesquisa dos Cafés do Brasil (4. : 2005 : Londrina, PR). Anais. Brasília, D.F. : Embrapa Café, 2005.

Disponível em: < <http://www.sbicafe.ufv.br/handle/10820/1508> > Acesso em: 12/02/2012.

ESPADA Rodolfo Jr., Armando APAN, Kevin MCDOUGALL (2014). Spatial modelling of natural disaster risk reduction policies with Markov decision processes. In: Applied Geography 53 (2014) 284-298.

Disponível em:< <http://dx.doi.org/10.1016/j.apgeog.2014.06.021>> Acesso em:28/10/2014.

FISHER Aiden, David A. GREEN , Andrew V. METCALFE , Kunle AKANDE (2014). First-passage criteria for the operation of reservoirs. In: Journal of Hydrology 519 (2014) 1836-1847.

Disponível em:< <http://dx.doi.org/10.1016/j.jhydrol.2014.09.061>> Acesso em: 28/10/2014.

GAITCHOUK L. Galtchouka, S. PERGAMENSHCHIKOV. (2014). Geometric ergodicity for classes of homogeneous Markov chains. In: Stochastic Processes and their Applications 124 (2014) 3362-3391.

Disponível em:< <http://dx.doi.org/10.1016/j.spa.2014.05.002>> Acesso em:28/10/2014.

GEORGIUO Nicholas, Andrew R. WADE. (2014). Non-homogeneous random walks on a semi-finite strip. In: Stochastic processes and their applications. 124 (2014) 3179-3205.

Disponível em:< <http://dx.doi.org/10.1016/j.spa.2014.05.005>> Acesso em:28/10/2014.

GONG W. Wenfeng, Li YUAN, Wenyi FAN, Philip STOTT. (2015). Analysis and simulation of land use spatial pattern in Harbin prefecture based on trajectories and cellular automata – Markov modelling. In: International Journal of Applied Earth Observation and Geoinformation 34 (2015) 207-216. Elsevier.

Disponível em:< <http://dx.doi.org/10.1016/j.jag.2014.07.005> > Acesso em:28/10/2014.

JANSSENS Davy, Geert WETS, Tom BRIJS, Koen VANHOOF (2005). The development of an adapted Markov chain modelling heuristic and simulation framework in the context of transportation research. In: Elsevier - Expert Systems with Applications 28 (2002) pg105-117.

Disponível em:< <http://dx.doi:10.1016/j.eswa.2004.08.004>> Acesso em:26/03/2013.

KELLER FILHO, Thadeu , Jurandir ZULLO JUNIOR e Paulo Roberto Schubnell DE REZENDE LIMA (2006). Analysis of the transition between dry and wet days through third-order Markov chains, Brazilian agriculture research, Brasília, v.41, n.9, p.1341-1349.

Disponível em: <<http://dx.doi.org/10.1590/S0100-204X2006000900001>> Acesso em: 26/03/2013.

KIJIMA, M. (1997). Markov Process for Stochastic Modeling. *chapter 1*. (ed. Chapman & Hall), University of Cambridge, London.

KOURITZIN Michael A. , Fraser NEWTON, Biao WU. (2014) A graph theoretic approach to simulation and classification. Computational Statistics and Data Analysis 70 (2014) 281-294.

Disponível em:< <http://dx.doi.org/10.1016/j.csda.2013.09.026>> Acesso em:28/10/2014.

KYRIAKIDIS, Epaminondas G. (1993). A Markov decision algorithm for optimal pest control through uniform catastrophes. In: *European Journal of Operational Research*, 64, 38-44.

Disponível em:< [10.1016/0377-2217\(93\)90006-9](http://dx.doi.org/10.1016/0377-2217(93)90006-9) > Acesso em:26/03/2013.

KYRIAKIDIS, Epaminondas G. (1995). Optimal pest control through the introduction of a predator, In: *European Journal of Operational Research*, 81, p.357-363.

Disponível em:< [10.1016/0377-2217\(93\)E0310-T](http://dx.doi.org/10.1016/0377-2217(93)E0310-T) > Acesso em:26/03/2013.

KYRIAKIDIS, Epaminondas G. (2011). Explicit Solution of Average-Cost Optimality Equation for a Pest-Control Problem., In: *Advances in decision sciences, Volume 2011, Article ID 917812*, 11 paginas.doi:10.1155/2011/617812.

Disponível em:< <http://dx.doi.org/10.1155/2011/617812>> Acesso em:26/03/2013.

LANDIM C. (2014). A topology for limits of Markov chains. In: Stochastic Processes and their Applications (2014).

Disponível em:<<http://dx.doi.org/10.1016/j.spa.2014.08.011>> Acesso em:28/10/2014.

LARSEN Christian, Marcel TURKENSTEEN. (2014). A vendor managed inventory model using continuous approximations for route length estimates and Markov chain modeling for cost estimates. In: Int. J. Production Economics (2014).

Disponível em: <<http://dx.doi.org/10.1016/j.ijpe.2014.08.001>> Acesso em:28/10/2014.

LUO Dong, Wenting ZHANG. (2014). A comparison of Markov model-based methods for predicting the ecosystem service value of land use in Wuhan, central China. In: Ecosystem Services 7 (2014) 57-65.

Disponível em:< <http://dx.doi.org/10.1016/j.ecoser.2013.11.001>> Acesso em:28/10/2014.

MAO Yong-Hua, Yan-Hong SONG. (2014). On geometric transience for discrete-time Markov chains. In: Stochastic Processes and their Applications 124 (2014) 1648-1678.

Disponível em:<<http://dx.doi.org/10.1016/j.spa.2013.12.012>> Acesso em:28/10/2014.

MASKAWA Jun-ichi (2003). Multivariate Markov chain modeling for stock markets. In: Physica A 324 (2003) 317 – 322.

Disponível em:<[http://dx.doi:10.1016/S0378-4371\(02\)01868-X](http://dx.doi:10.1016/S0378-4371(02)01868-X)> Acesso em:28/10/2014.

MENSHIKOV Mikhail, Dimitri PETRITIS. (2014). Explosion, implosion, and moments of passage times for continuous-time Markov chains: A semimartingale approach. In: Stochastic Processes and their Applications 124 (2014). 2388-2414.

Disponível em:<<http://dx.doi.org/10.1016/j.spa.2014.03.001>> Acesso em:28/10/2014.

NOGUEIRA, F. Modelagem e Simulação – Cadeias de Markov. 2009. 40p. Notas de aula.

Disponível em:< <http://www.ufjf.br/epd042/files/2009/02/cadeiaMarkov1.pdf>> Acesso em: 28/09/2012.

NORRIS, J. R. (1997). Markov Chains. Introduction. (ed. Cambridge University Press), University of Cambridge, London.

NUNES, José Luis da Silva, Eng. Agrº, Dr. em Fitotecnia . Site da Internet:

<http://www.agrolink.com.br/georreferenciamento/AgriculturaPrecisao.aspx> - acessado em 18/12/2014 8h30min

STRIGUL Nikolay , Ionut FLORESCU, Alicia R. WELDEN, Fabian MICHALCZEWSKI (2012). Modelling of Forest stand dynamics using Markov Chains. In: Environmental Modelling & Software 31 (2012) 64-75.

Disponível em:<<http://dx.doi.org/10.1016/j.envsift.2011.12.004>> Acesso em:28/10/2014.

SUN Dazhi , Xuqian LI (2010). Application of Markov Chain Model on Environmental Fate of Phenanthrene in Soil and Groundwater. In: *Procedia Environmental Sciences* 2 (2010) 814-823.

Disponível em:< [10.1016/j.proenv.2010.10.092](http://dx.doi.org/10.1016/j.proenv.2010.10.092)> Acesso em:28/10/2014.

THOMPSON, Dan K. , WADDINGTON, James M., (2014). A markov chain method for simulating bulk density profiles in boreal peatlands. In: *Geoderma* 232-234 (2014) 123-129. Elsevier.

Disponível em:< <http://dx.doi.org/10.1016/j.geoderma.2014.04.032>> Acesso em: 28/10/2014.

TILOKAVICHAI Varanya, Peraphon SOPHATSATHIT, Achara CHANDRACHAI (2012). Innovative Logistics Management under Uncertainty using Markov Model. In: *The International Institute for Science, Technology and Education (IISTE)*, Vol. 2, No.5.

Disponível em:< <http://www.iiste.org/Journals/index.php/IKM/article/view/2244/2256>> Acesso em:26/03/2013.

VINATIER Fabrice , Françoise LESCOURRETB, Pierre-François DUYCK, Philippe TIXIER (2012). From IBM to IPM: Using individual-based models to design the spatial arrangement of traps and crops in integrated pest management strategies. In: *Agriculture, Ecosystems and Environment* 146 (2012) 52-59.

Disponível em:< [10.1016/j.agee.2011.10.005](http://dx.doi.org/10.1016/j.agee.2011.10.005)> Acesso em:28/10/2014.

WILCOX William , Philip A. HORVATH, Stanley E. GRIFFIS, Chad W. AUTRY, (2011). A Markov model of liquidity effects in reverse logistics processes: The effects of random volume and passage.In. *J. Production Economics* 129 (2011) pg 86-101.

Disponível em:< <http://dx.doi.org/10.1016/j.jipe.2010.09.005> > Acesso em:26/03/2013.

YANG Xin, Xin-Qi ZHENG, Rui CHEN. (2014). A land use change model: Integrating landscape pattern indexes and Markov-CA. In: *Ecological Modelling* 283 (2014) 1-7. Elsevier.

Disponível em:< <http://dx.doi.org/10.1016/j.ecolmodel.2014.03.011>> Acesso em:28/10/2014.

ZUBIZARRETA A., A. GERENDIAINA, M.P. FERNÁNDEZ.(2014). Relative branch size in branch clusters through a Markovian process. In: *Ecological Modelling* 273 (2014) 210-219. Elsevier.

Disponível em:< <http://dx.doi.org/10.1016/j.ecolmodel.2013.11.012>> Acesso em:28/10/2014.

APÊNDICE A

CÓDIGO PARA O TRATAMENTO DOS DADOS BRUTOS

O seguinte código foi desenvolvido em MatLab® para a rotina do tratamento dos dados brutos:

```
clc;clear all; close all;
% leitura de dados 2012/2013
tic
disp('Reading 2012/2013 data...');
files={'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 09 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 09 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 10 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 10 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 11 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 11 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2012 12 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 01 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 01 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 02 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 02 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 03 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 03 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 04 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 04 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 05 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 05 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 06 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 06 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 07 1'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 07 2'
'CONTROLE DE PRAGAS 2012 2013/MANEJO ECOLOGICO 2013 08 1'
};% lista de todos arquivos de dados brutos

% inicialização da matriz de dados tratados
data1213=zeros(106,7,size(files,1));

for ii=1:size(files,1)
[num,txt,row] = xlsread(cell2mat(files(ii,:)), 'Demonstrativo');
%posiciona-se na planilha demonstrativo para leitura

if size(num,1)>112
%Elimina as áreas com dados incompletos das planilhas mais
recentes
num([32,55,102:107,113:size(num,1)],:)=[];
```

```

else
    num([32,55,102:107],:)=[]; %Elimina as áreas com dados incompletos
    das planilhas mais antigas

end

data1213(1:3,1:7,ii)=num(1:3,1:7); %Identifica os dados do ano 2013

data1213(6:size(num,1)+2,1:7,ii)=num(4:size(num,1),1:7);
end
disp('done!')
toc
clear files num raw txt ii

% reading data 2011/2012
disp('Reading 2011/2012 data...');
files={'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 09 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 09 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 10 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 10 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 11 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 11 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2011 12 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 01 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 01 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 02 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 02 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 03 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 03 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 04 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 04 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 05 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 05 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 06 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 06 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 07 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 07 2'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 08 1'
'CONTROLE DE PRAGAS 2011 2012/MANEJO ECOLOGICO 2012 08 2'
};

% inicialização das variáveis
data1112=zeros(106,7,size(files,1));

% leitura do arquivo de dados excel
for ii=1:size(files,1)
    [num,txt,raw] = xlsread(cell2mat(files(ii,:)), 'Demonstrativo');
    data1112(:, :, ii)=num(:,1:7);
end

% string com o nome das pragas
diseases=(txt(3,4:10));

clear files num txt raw ii
disp('done!')
toc
% lendo os dados 2010/2011

```

```

disp('Reading 2010/2011 data...');
files={
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 09 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 09 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 10 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 10 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 11 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 11 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2010 12 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 01 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 01 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 02 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 02 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 03 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 03 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 04 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 04 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 05 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 05 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 06 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 06 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 07 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 07 2'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 08 1'
    'CONTROLE DE PRAGAS 2010 2011/MANEJO ECOLOGICO 2011 08 2'
};

% matriz de dados
data1011=zeros(106,7,size(files,1));

for ii=1:size(files,1)
    [num,txt,raw] = xlsread(cell2mat(files(ii,:)), 'Demonstrativo');

    % eliminando os dados indesejaveis
    if size(num,1)>106
        num(102:106,:)=[];
    end
    data1011(1:size(num,1),1:7,ii)=num(:,1:7);
end

clear files num txt raw ii
disp('done!')
save('data.mat')
disp('data saved!')
toc

```

CÓDIGO EM MatLab® PARA O TRATAMENTO DA PLANILHA DE PULVERIZAÇÃO

O seguinte código foi desenvolvido em MatLab® para a rotina do tratamento dos dados de pulverização:

```

clc;clear all;close all;
files={'CONTROLE DE PRAGAS 2010 2011/Pulverizações Realizadas'
      'CONTROLE DE PRAGAS 2011 2012/Pulverizações Realizadas2011-2012'
      'CONTROLE DE PRAGAS 2012 2013/Pulverizações Realizadas2012-2013'
      };
% Leitor
[num1,txt1,raw1] = xlsread(cell2mat(files(1,:)),'read');
[num2,txt2,raw2] = xlsread(cell2mat(files(2,:)),'read');
[num3,txt3,raw3] = xlsread(cell2mat(files(3,:)),'read');

clear files

% Tratamento dos dados
% Eliminando linhas indesejáveis - talhões não analisados
%      colunas indesejáveis - quinzena não analisada (repetição
agosto)
Pulv1011=num1;
Pulv1011(:, [7 8 9])=[];
Pulv1011([9 57 104:108],:)=[];

clear num1 txt1 raw1

Pulv1112=num2;
Pulv1112(:, [1 2])=[];
Pulv1112([34 57 104:109],:)=[];

clear num2 txt2 raw2

Pulv1213=num3;
Pulv1213(:, [1 2 25])=[];
Pulv1213([34 57 104:109 115:120],:)=[];

clear num3 txt3 raw3

```

APÊNDICE B

CÓDIGO DESENVOLVIDO EM MatLab® PARA A OBTENÇÃO DAS MATRIZES DE TRANSIÇÃO

O seguinte código foi desenvolvido em MatLab® para a obtenção das matrizes de transição de dois estados:

Algoritmo principal para a definição de dois estados

```
clc;clear all;close all;
% Carrega a matriz de dados tratados
load('data');
% Definição dos limites da baixa incidência
limits=[5 0 10 5];
% ===== Criação da Matriz de Transição =====
%Identificação dos estados de baixa e alta incidência
for kk=1:length(limits);
    idx1011(:,kk,:)= data1011(:,kk,:)>=limits(kk);
    idx1112(:,kk,:)= data1112(:,kk,:)>=limits(kk);
    idx1213(:,kk,:)= data1213(:,kk,:)>=limits(kk);
end
clear kk
% Gera a matriz de transição
T1011_BMin=matrix_gen(idx1011,1);
T1112_BMin=matrix_gen(idx1112,1);
T1213_BMin=matrix_gen(idx1213,1);
T1011_AcVerm=matrix_gen(idx1011,3);
T1112_AcVerm=matrix_gen(idx1112,3);
T1213_AcVerm=matrix_gen(idx1213,3);
T1011_Ferr=matrix_gen(idx1011,4);
T1112_Ferr=matrix_gen(idx1112,4);
T1213_Ferr=matrix_gen(idx1213,4);
% Agrupa por praga
T_BMin(:,:,1:106)=T1011_BMin;
T_BMin(:,:,107:212)=T1112_BMin;
T_BMin(:,:,213:318)=T1213_BMin;
T_AcVerm(:,:,1:106)=T1011_AcVerm;
T_AcVerm(:,:,107:212)=T1112_AcVerm;
T_AcVerm(:,:,213:318)=T1213_AcVerm;
T_Ferr(:,:,1:106)=T1011_Ferr;
T_Ferr(:,:,107:212)=T1112_Ferr;
T_Ferr(:,:,213:318)=T1213_Ferr;

clear aux
```

Function matrix_gen: Função que gera a matriz de transição para o modelo de dois estados a partir da matriz de dados referente aos estados do sistema ao longo do tempo

```
function [T]=matrix_gen(idx,jj);

% Função que gera a matriz de incidência T baseada no vetor idx referente à
% praga da coluna 'jj'
%
% O sistema possui dois estados possíveis 1 - para alta incidência e 0 para
% baixa incidência. Desta forma as transições (posições da matriz) são
% alocadas de acordo com a descrição a seguir:
%
% Posição 11 - Transição de baixa incidência para baixa incidência
% Posição 12 - Transição de baixa incidência para alta incidência
% Posição 21 - Transição de alta incidência para baixa incidência
% Posição 22 - Transição de alta incidência para alta incidência
%
% A matriz T possui dimensões 2 x 2 x n
%
% Onde n é o número de áreas analisadas. n = size(idx,1)

T=zeros(2,2,size(idx,1));

for ii=1:size(idx,1)

    a=idx(ii,jj,:);
    t00 = zeros(size(a,3));
    t01 = zeros(size(a,3));
    t10 = zeros(size(a,3));
    t11 = zeros(size(a,3));

    for kk=2:22

        if (a(1,1,kk)==0)&&a(1,1,kk-1)==0
            t00(ii)=t00(ii)+1;
        end

        if (a(1,1,kk)==0)&&a(1,1,kk-1)==1
            t01(ii)=t01(ii)+1;
        end

        if (a(1,1,kk)==1)&&a(1,1,kk-1)==0
            t10(ii)=t10(ii)+1;
        end

        if (a(1,1,kk)==1)&&a(1,1,kk-1)==1
            t11(ii)=t11(ii)+1;
        end

        T(1:2,1:2,ii)=[t00(ii) t01(ii);t10(ii) t11(ii)];

    end

end

end
```

O seguinte código foi desenvolvido em MatLab® para a obtenção das matrizes de transição de quatro estados:

Algoritmo principal para a definição de quatro estados

```

clc;clear all;close all;
files={'CONTROLE DE PRAGAS 2010 2011/Pulverizações Realizadas'
      'CONTROLE DE PRAGAS 2011 2012/Pulverizações Realizadas2011-2012'
      'CONTROLE DE PRAGAS 2012 2013/Pulverizações Realizadas2012-2013'
      };
% Leitor
[num1,txt1,raw1] = xlsread(cell2mat(files(1,:)), 'read');
[num2,txt2,raw2] = xlsread(cell2mat(files(2,:)), 'read');
[num3,txt3,raw3] = xlsread(cell2mat(files(3,:)), 'read');

clear files

% Tratamento dos dados
% Eliminando linhas indesejáveis - talhões não analisados
%      colunas indesejáveis - quinzena não analisada
Pulv1011=num1;
Pulv1011(:, [7 8 9])=[];
Pulv1011([9 57 104:108],:)=[];

clear num1 txt1 raw1

Pulv1112=num2;
Pulv1112(:, [1 2])=[];
Pulv1112([34 57 104:109],:)=[];

clear num2 txt2 raw2

Pulv1213=num3;
Pulv1213(:, [1 2 25])=[];
Pulv1213([34 57 104:109 115:120],:)=[];

clear num3 txt3 raw3

% Carregando os dados ...
load('data');

% Definindo os limites de baixa incidência
limits=[5 0 10 5];

% ===== Criando a matriz de transição =====
for kk=1:length(limits);
    idx1011(:,kk,:)= data1011(:,kk,:)>=limits(kk);
    idx1112(:,kk,:)= data1112(:,kk,:)>=limits(kk);
    idx1213(:,kk,:)= data1213(:,kk,:)>=limits(kk);
end
clear kk

% Identificando o estado pulverizado
idx1011=double(idx1011);

```

```

idx1112=double (idx1112);
idx1213=double (idx1213);

%pause

for ii=1:size(Pulv1011,1)
    for jj=1:size(Pulv1011,2)
        for kk=1:size(idx1011,2)
            % pulverizado e baixa incidencia
            if Pulv1011(ii,jj)==1&&idx1011(ii,kk,jj)==0
                idx1011(ii,:,jj)=2;
            end
            %pulverizado e alta incidencia
            if Pulv1011(ii,jj)==1&&idx1011(ii,kk,jj)==1
                idx1011(ii,:,jj)=3;
            end
        end
    end
end

for ii=1:size(Pulv1112,1)
    for jj=1:size(Pulv1112,2)
        for kk=1:size(idx1112,2)
            % pulverizado e baixa incidencia
            if Pulv1112(ii,jj)==1&&idx1112(ii,kk,jj)==0
                idx1112(ii,:,jj)=2;
            end
            %pulverizado e alta incidencia
            if Pulv1112(ii,jj)==1&&idx1112(ii,kk,jj)==1
                idx1112(ii,:,jj)=3;
            end
        end
    end
end

for ii=1:size(Pulv1213,1)
    for jj=1:size(Pulv1213,2)
        for kk=1:size(idx1213,2)
            % pulverizado e baixa incidencia
            if Pulv1213(ii,jj)==1&&idx1213(ii,kk,jj)==0
                idx1213(ii,:,jj)=2;
            end
            %pulverizado e alta incidencia
            if Pulv1213(ii,jj)==1&&idx1213(ii,kk,jj)==1
                idx1213(ii,:,jj)=3;
            end
        end
    end
end

% Gerando as matrizes de transição
% Bicho Mineiro Anual_____

```



```

T1011_BMin=matrix_gen3(idx1011,1,2,22);
for k=1:size (T1011_BMin,3)
    T1011_BMinP(:, :,k)= transf_em_porcentagem(T1011_BMin(:, :,k));
end
%clear T1011_BMin

T1112_BMin=matrix_gen3(idx1112,1,2,23);
for k=1:size (T1112_BMin,3)
    T1112_BMinP(:, :,k)= transf_em_porcentagem(T1112_BMin(:, :,k));
end
%clear T1112_BMin

T1213_BMin=matrix_gen3(idx1213,1,2,22);
for k=1:size (T1213_BMin,3)
    T1213_BMinP(:, :,k)= transf_em_porcentagem(T1213_BMin(:, :,k));
end
%clear T1213_BMin

% Acaro Vermelho Anual _____
T1011_AcVerm=matrix_gen3(idx1011,3,2,22);
for k=1:size (T1011_AcVerm,3)
    T1011_AcVermP(:, :,k)= transf_em_porcentagem(T1011_AcVerm(:, :,k));
end
%clear T1011_AcVerm

T1112_AcVerm=matrix_gen3(idx1112,3,2,23);
for k=1:size (T1112_AcVerm,3)
    T1112_AcVermP(:, :,k)= transf_em_porcentagem(T1112_AcVerm(:, :,k));
end
%clear T1112_AcVerm

T1213_AcVerm=matrix_gen3(idx1213,3,2,22);
for k=1:size (T1213_AcVerm,3)
    T1213_AcVermP(:, :,k)= transf_em_porcentagem(T1213_AcVerm(:, :,k));
end
%clear T1213_AcVerm

% Ferrugem Anual _____
T1011_Ferr=matrix_gen3(idx1011,4,2,22);
for k=1:size (T1011_Ferr,3)
    T1011_FerrP(:, :,k)= transf_em_porcentagem(T1011_Ferr(:, :,k));
end
%clear T1011_Ferr

T1112_Ferr=matrix_gen3(idx1112,4,2,23);
for k=1:size (T1112_Ferr,3)
    T1112_FerrP(:, :,k)= transf_em_porcentagem(T1112_Ferr(:, :,k));
end
%clear T1112_Ferr

T1213_Ferr=matrix_gen3(idx1213,4,2,22);
for k=1:size (T1213_Ferr,3)
    T1213_FerrP(:, :,k)= transf_em_porcentagem(T1213_Ferr(:, :,k));
end
%clear T1213_Ferr

save results.mat

```

Function matrix_gen: Função que gera a matriz de transição para o modelo de quatro estados a partir da matriz de dados referente aos estados do sistema ao longo do tempo

```
function [T]=matrix_gen3(idx,jj,inf,sup);

% Função que gera a matriz de transição 'T' baseado no vetor idx referente
% à praga na coluna 'jj'
%
% A matriz T retorna valores inteiros referentes a frequência absoluta das
% transições
%
% A matriz possui transições referentes a quatro estados possíveis:
%
% O estado 1 - Baixa incidência e não pulverizado
% O estado 2 - Alta incidência e não pulverizado
% O estado 3 - Baixa incidência e pulverizado
% O estado 4 - Alta incidência e pulverizado
%
% A matriz T possui 3 dimensões 4 x 4 x n
%
% Onde n é o número de áreas: n = size(idx,1)
% inf = o intervalo inicial
% sup = o intervalo final
% jj se refere à praga:
%
%           1 - Praga 1
%           3 - Praga 2
%           4 - Praga 3
%
% idx se refere a matriz de estados com os estados alta e baixa incidência
% identificados. Restando a identificação da pulverização(database)

T=zeros(4,4,size(idx,1));

for ii=1:size(idx,1)

    a=idx(ii,jj,:);
    t00 = zeros(size(a,3));
    t01 = zeros(size(a,3));
    t02 = zeros(size(a,3));
    t03 = zeros(size(a,3));
    t10 = zeros(size(a,3));
    t11 = zeros(size(a,3));
    t12 = zeros(size(a,3));
    t13 = zeros(size(a,3));
    t20 = zeros(size(a,3));
    t21 = zeros(size(a,3));
    t22 = zeros(size(a,3));
    t23 = zeros(size(a,3));
    t30 = zeros(size(a,3));
    t31 = zeros(size(a,3));
    t32 = zeros(size(a,3));
    t33 = zeros(size(a,3));

    for kk=inf:sup

        if (a(1,1,kk)==0)&&a(1,1,kk-1)==0
            t00(ii)=t00(ii)+1;
        end
    end
end
```

```
if (a(1,1,kk)==0) &&a(1,1,kk-1)==1
    t01(ii)=t01(ii)+1;
end

if (a(1,1,kk)==0) &&a(1,1,kk-1)==2
    t02(ii)=t02(ii)+1;
end

if (a(1,1,kk)==0) &&a(1,1,kk-1)==3
    t03(ii)=t03(ii)+1;
end

if (a(1,1,kk)==1) &&a(1,1,kk-1)==0
    t10(ii)=t10(ii)+1;
end

if (a(1,1,kk)==1) &&a(1,1,kk-1)==1
    t11(ii)=t11(ii)+1;
end

if (a(1,1,kk)==1) &&a(1,1,kk-1)==2
    t12(ii)=t12(ii)+1;
end

if (a(1,1,kk)==1) &&a(1,1,kk-1)==3
    t13(ii)=t13(ii)+1;
end

if (a(1,1,kk)==2) &&a(1,1,kk-1)==0
    t20(ii)=t20(ii)+1;
end

if (a(1,1,kk)==2) &&a(1,1,kk-1)==1
    t21(ii)=t21(ii)+1;
end

if (a(1,1,kk)==2) &&a(1,1,kk-1)==2
    t22(ii)=t22(ii)+1;
end

if (a(1,1,kk)==2) &&a(1,1,kk-1)==3
    t23(ii)=t23(ii)+1;
end

if (a(1,1,kk)==3) &&a(1,1,kk-1)==0
    t30(ii)=t30(ii)+1;
end

if (a(1,1,kk)==3) &&a(1,1,kk-1)==1
    t31(ii)=t31(ii)+1;
end

if (a(1,1,kk)==3) &&a(1,1,kk-1)==2
    t32(ii)=t32(ii)+1;
end

if (a(1,1,kk)==3) &&a(1,1,kk-1)==3
```

```

        t33(ii)=t33(ii)+1;
    end

    T(1:4,1:4,ii)=[t00(ii) t01(ii) t02(ii) t03(ii);
                  t10(ii) t11(ii) t12(ii) t13(ii);
                  t20(ii) t21(ii) t22(ii) t23(ii);
                  t30(ii) t31(ii) t32(ii) t33(ii);
                  ];

    end

end

end

```

Function transf em porcentagem: Função que transforma os valores absolutos da matriz de transição em valores relativos respeitando a propriedade de matrizes estocásticas

```

function T= transf_em_porcentagem(M)

% Esta função realiza a transformação de uma matriz com valores absolutos
% de ocorrência em uma matriz de transição de uma cadeia de Markov
% respeitando a propriedade de matrizes estocásticas.
% As matrizes de transição têm o somatório da linha unitário.

[lin col]=size(M); %controle do tamanho da matriz

T=zeros(lin,col); % inicializa a matriz de retorno

for ii=1:lin
    soma=sum(M,2);
    if soma(ii)==0
        T(ii,:)=zeros(1,col);
    else
        T(ii,:)=M(ii,:)/soma(ii);
    end
end

end
clear soma ii lin col;
end

```

APÊNDICE C

MATRIZES DE TRANSIÇÃO ENCONTRADAS

<u>T1011 – Praga 1</u>				0.6667	0	0.2667	0.0667
val(:, :, 1) =				0	0	0	0
				1.0000	0	0	0
				0	0	1.0000	0
0.6875	0	0.3125	0				
0	0	0	0				
1.0000	0	0	0				
0	0	0	0				
val(:, :, 2) =				0.6667	0	0.2667	0.0667
				0	0	0	0
				1.0000	0	0	0
				0	0	1.0000	0
0.6875	0	0.2500	0.0625				
0	0	0	0				
1.0000	0	0	0				
1.0000	0	0	0				
val(:, :, 3) =				0.6667	0	0.3333	0
				0	0	0	0
				0.8333	0	0.1667	0
				0	0	0	0
0.6875	0	0.3125	0				
0	0	0	0				
1.0000	0	0	0				
0	0	0	0				
val(:, :, 4) =				0.6667	0	0.2667	0.0667
				0	0	0	0
				1.0000	0	0	0
				0	0	1.0000	0
0.7333	0	0.2000	0.0667				
0	0	0	0				
0.6000	0	0.4000	0				
1.0000	0	0	0				
val(:, :, 5) =				0.7500	0	0.2500	0
				0	0	0	0
				0.3333	0	0.6667	0
				0	0	0	0
0.7333	0	0.2000	0.0667				
0	0	0	0				
0.6000	0	0.4000	0				
1.0000	0	0	0				
val(:, :, 6) =				0.7857	0	0.1429	0.0714
				0	0	0	0
				0.5000	0	0.5000	0
				0	0	1.0000	0
0.7647	0.0588	0.1765	0				
1.0000	0	0	0				
1.0000	0	0	0				
0	0	0	0				
val(:, :, 7) =				0.7857	0	0.2143	0
				0	0	0	0
				0.4286	0	0.5714	0
				0	0	0	0
0.6154	0	0.3846	0				
0	0	0	0				
0.6250	0	0.3750	0				
0	0	0	0				
val(:, :, 8) =				0.7857	0	0.2143	0
				0	0	0	0
				0.4286	0	0.5714	0
				0	0	0	0

```

                                0.6154      0      0.3077      0.0769
val(:, :, 16) =                0          0          0          0
                                0.5714      0      0.4286      0
                                1.0000      0          0          0
    0.6429      0      0.3571      0
      0          0          0          0
    0.7143      0      0.2857      0
      0          0          0          0

                                0.6154      0      0.3846      0
val(:, :, 17) =                0          0          0          0
                                0.6250      0      0.3750      0
                                0          0          0          0
    0.6667      0      0.2667      0.0667
      0          0          0          0
    0.8000      0      0.2000      0
    1.0000      0          0          0

                                0.6154      0      0.3077      0.0769
val(:, :, 18) =                0          0          0          0
                                0.5714      0      0.4286      0
                                1.0000      0          0          0
    0.6667      0      0.2667      0.0667
      0          0          0          0
    0.8000      0      0.2000      0
    1.0000      0          0          0

                                0.6154      0      0.3077      0.0769
val(:, :, 19) =                0          0          0          0
                                0.5714      0      0.4286      0
                                1.0000      0          0          0
    0.6667      0      0.2667      0.0667
      0          0          0          0
    0.8000      0      0.2000      0
    1.0000      0          0          0

                                0.6154      0      0.3846      0
val(:, :, 20) =                0          0          0          0
                                0.6250      0      0.3750      0
                                0          0          0          0
    0.5714      0      0.3571      0.0714
      0          0          0          0
    0.8333      0      0.1667      0
    1.0000      0          0          0

                                0.6154      0      0.3077      0.0769
val(:, :, 21) =                0          0          0          0
                                0.4286      0      0.5714      0
                                1.0000      0          0          0
    0.5714      0      0.4286      0
      0          0          0          0
    0.8571      0      0.1429      0
      0          0          0          0

                                0.6667      0      0.2667      0.0667
val(:, :, 22) =                0          0          0          0
                                0.6000      0      0.4000      0
                                1.0000      0          0          0
    0.5714      0      0.3571      0.0714
      0          0          0          0
    0.8333      0      0.1667      0
    1.0000      0          0          0

                                0.6875      0      0.1875      0.1250
val(:, :, 23) =                0          0          0          0
                                1.0000      0          0          0
                                1.0000      0          0          0
    0.5833      0      0.3333      0.0833
      0          0          0          0
    0.5000      0      0.5000      0
    1.0000      0          0          0

                                0.7647      0      0.1765      0.0588
val(:, :, 24) =                0          0          0          0
                                1.0000      0          0          0
                                1.0000      0          0          0
    0.6154      0      0.3077      0.0769
      0          0          0          0
    0.5714      0      0.4286      0
    1.0000      0          0          0

                                0.7647      0      0.1765      0.0588
val(:, :, 25) =                0          0          0          0

```

```

1.0000      0      0      0
1.0000      0      0      0
val(:, :, 35) =
0.7647      0      0.2353      0
0           0           0           0
1.0000      0           0           0
0           0           0           0
val(:, :, 36) =
0.6667      0      0.3333      0
1.0000      0           0           0
1.0000      0           0           0
0           0           0           0
val(:, :, 37) =
0.6667      0      0.3333      0
1.0000      0           0           0
1.0000      0           0           0
0           0           0           0
val(:, :, 38) =
0.6667      0      0.2667      0.0667
1.0000      0           0           0
1.0000      0           0           0
1.0000      0           0           0
val(:, :, 39) =
0.6667      0      0.2667      0.0667
1.0000      0           0           0
1.0000      0           0           0
1.0000      0           0           0
val(:, :, 40) =
0.6429      0      0.2857      0.0714
1.0000      0           0           0
0.8000      0      0.2000      0
1.0000      0           0           0
val(:, :, 41) =
0.6667      0      0.3333      0
1.0000      0           0           0
1.0000      0           0           0
0           0           0           0
val(:, :, 42) =
0.6667      0      0.2667      0.0667
1.0000      0           0           0
1.0000      0           0           0
1.0000      0           0           0
val(:, :, 43) =
0.6667      0      0.3333      0
1.0000      0           0           0
1.0000      0           0           0
0           0           0           0
val(:, :, 44) =
0.9444      0.0556      0      0
0.6667      0.3333      0      0
0           0           0      0
0           0           0      0
val(:, :, 45) =
0.6154      0      0.3077      0.0769
0           0           0           0
0.6667      0      0.1667      0.1667
0.5000      0      0.5000      0
val(:, :, 46) =
0.6429      0      0.2857      0.0714
0           0           0           0
0.8000      0           0           0.2000
0.5000      0      0.5000      0
val(:, :, 47) =
0.6429      0      0.2143      0.1429
0           0           0           0
1.0000      0           0           0
0.3333      0      0.3333      0.3333
val(:, :, 48) =
0.6429      0      0.2857      0.0714
0           0           0           0
0.8000      0      0.2000      0
0.5000      0           0           0.5000
val(:, :, 49) =
0.6429      0      0.2857      0.0714
0           0           0           0
0.8000      0      0.2000      0
0.5000      0           0           0.5000
val(:, :, 50) =
0.7143      0.0714      0.1429      0.0714
0           0           0           1.0000
1.0000      0           0           0
0.3333      0      0.3333      0.3333
val(:, :, 51) =
0.6667      0.1111      0.2222      0
0.1667      0.6667      0.1667      0
0.6000      0           0.4000      0
0           1.0000      0           0
val(:, :, 52) =
0.5000      0      0.4167      0.0833
1.0000      0           0           0
0.8333      0           0           0.1667
0           0.5000      0.5000      0
val(:, :, 53) =

```

0.5000	0	0.5000	0	0	0	0	0
1.0000	0	0	0				
0.7143	0	0.1429	0.1429				
0	1.0000	0	0	val(:, :, 63) =			
				0.7647	0	0.1765	0.0588
val(:, :, 54) =				0	0	0	0
				1.0000	0	0	0
0.4545	0	0.5455	0	1.0000	0	0	0
1.0000	0	0	0				
0.6250	0	0.2500	0.1250				
0	1.0000	0	0	val(:, :, 64) =			
				0.7647	0	0.1765	0.0588
val(:, :, 55) =				0	0	0	0
				1.0000	0	0	0
0.9412	0.0588	0	0	1.0000	0	0	0
0.6667	0.3333	0	0				
0	0	0	0				
0	1.0000	0	0	val(:, :, 65) =			
				0.6667	0	0.3333	0
val(:, :, 56) =				0	0	0	0
				0.8333	0	0.1667	0
0.6667	0	0.3333	0	0	0	0	0
0.3333	0.6667	0	0				
0.8000	0	0.2000	0				
0	1.0000	0	0	val(:, :, 66) =			
				0.6667	0	0.3333	0
val(:, :, 57) =				0	0	0	0
				0.8333	0	0.1667	0
0.4545	0	0.5455	0	0	0	0	0
1.0000	0	0	0				
0.5556	0.1111	0.3333	0				
0	0	0	0	val(:, :, 67) =			
				0.8235	0	0.1765	0
val(:, :, 58) =				0	0	0	0
				0.7500	0	0.2500	0
0.4545	0	0.4545	0.0909	0	0	0	0
1.0000	0	0	0				
0.6250	0.1250	0.2500	0				
0	0	1.0000	0	val(:, :, 68) =			
				0.7500	0	0.2500	0
val(:, :, 59) =				0	0	0	0
				0.8000	0	0.2000	0
0.4545	0	0.4545	0.0909	0	0	0	0
1.0000	0	0	0				
0.6250	0.1250	0.2500	0				
0	0	1.0000	0	val(:, :, 69) =			
				0.6923	0	0.2308	0.0769
val(:, :, 60) =				0	0	0	0
				0.5714	0	0.4286	0
0.6154	0.0769	0.3077	0	0	0	1.0000	0
1.0000	0	0	0				
0.6667	0	0.3333	0				
1.0000	0	0	0	val(:, :, 70) =			
				0.7143	0	0.2143	0.0714
val(:, :, 61) =				0	0	0	0
				0.5000	0	0.5000	0
0.6875	0	0.2500	0.0625	1.0000	0	0	0
0	0	0	0				
1.0000	0	0	0				
1.0000	0	0	0	val(:, :, 71) =			
				0.7143	0	0.2143	0.0714
val(:, :, 62) =				0	0	0	0
				0.5000	0	0.5000	0
0.7647	0	0.2353	0	1.0000	0	0	0
0	0	0	0				
1.0000	0	0	0				


```

val(:,:,72) =
    0.7143    0    0.2143    0.0714
    0.6667    0    0.3333    0
    0.0000    0    1.0000    0
    0.6000    0    0.4000    0
    1.0000    0    0.0000    0
val(:,:,82) =
    0.7333    0    0.2667    0
    0.6667    0    0.3333    0
    0.0000    0    0.0000    0
val(:,:,73) =
    0.7333    0    0.2000    0.0667
    0.8000    0    0.2000    0
    0.0000    0    1.0000    0
val(:,:,83) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,74) =
    0.7333    0    0.2000    0.0667
    0.8000    0    0.2000    0
    0.0000    0    1.0000    0
val(:,:,84) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,75) =
    0.7692    0    0.1538    0.0769
    0.2857    0    0.7143    0
    1.0000    0    0.0000    0
val(:,:,85) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,76) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,86) =
    0.6000    0    0.3333    0.0667
    0.8000    0    0.2000    0
    1.0000    0    0.0000    0
val(:,:,77) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,87) =
    0.6667    0    0.2222    0.1111
    0.4000    0.8000    0.2000    0
    0.5000    0    0.6000    0
    0.5000    0.5000    0.0000    0
val(:,:,78) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,88) =
    0.6667    0    0.3333    0
    0.5000    0.8000    0.2000    0
    0.5000    0    0.5000    0
    0.0000    1.0000    0.0000    0
val(:,:,79) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,89) =
    0.6000    0    0.3000    0.1000
    0.2500    0.7500    0.0000    0
    0.4000    0    0.6000    0
    0.5000    0.5000    0.0000    0
val(:,:,80) =
    0.7143    0    0.2143    0.0714
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0
val(:,:,90) =
    0.6000    0    0.3000    0.1000
    0.2500    0.7500    0.0000    0
    0.4000    0    0.6000    0
    0.5000    0.5000    0.0000    0
val(:,:,81) =
    0.7333    0    0.2000    0.0667
    0.5000    0    0.5000    0
    1.0000    0    0.0000    0

```

```

                                0.6923      0      0.2308      0.0769
val(:, :, 91) =                0          0          0          0
                                0.6667      0      0.3333      0
                                0          0      0.5000      0.5000
0.5556      0      0.4444      0
0          0.8000      0.2000      0
0.6667      0      0.3333      0
0          1.0000      0          0

                                0.6923      0      0.2308      0.0769
val(:, :, 92) =                0          0          0          0
                                0.6667      0      0.3333      0
                                0          0      0.5000      0.5000
0.7333      0      0.2000      0.0667
0          0          1.0000      0
0.7500      0.2500      0          0
1.0000      0          0          0

                                0.7500      0      0.1875      0.0625
val(:, :, 93) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          1.0000      0
0.7333      0      0.2667      0
0          0          1.0000      0
0.8000      0.2000      0          0
0          0          0          0

                                0.7500      0      0.1875      0.0625
val(:, :, 94) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          1.0000      0
0.7333      0      0.2667      0
0          0          1.0000      0
0.8000      0.2000      0          0
0          0          0          0

                                0.7500      0      0.1875      0.0625
val(:, :, 95) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          1.0000      0
0.7333      0      0.2667      0
0          0          1.0000      0
0.8000      0.2000      0          0
0          0          0          0

                                0.7500      0      0.1875      0.0625
val(:, :, 96) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          1.0000      0
0.7333      0      0.2667      0
0          0          1.0000      0
0.8000      0.2000      0          0
0          0          0          0

                                0.7500      0      0.1875      0.0625
val(:, :, 97) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          1.0000      0
0.5833      0      0.4167      0
0.5000      0.5000      0          0
0.7143      0      0.2857      0
0          0          0          0

                                0.6429      0      0.3571      0
val(:, :, 98) =                0.5000      0.5000      0          0
                                0.8000      0.2000      0          0
                                0          0          0          0
0.6923      0      0.2308      0.0769
0.5000      0.5000      0          0
0.6000      0      0.4000      0
1.0000      0          0          0

                                0.6875      0      0.2500      0.0625
val(:, :, 99) =                0          0          0          0
                                1.0000      0          0          0
                                1.0000      0          0          0
0.6923      0      0.2308      0.0769
0.5000      0.5000      0          0
0.6000      0      0.4000      0
1.0000      0          0          0

                                0.6000      0.0667      0.3333      0
val(:, :, 100) =

```

T1011 - Praga 2

val(:, :, 1) =

val(:, :, 2) =

val(:, :, 3) =

	0.4000	0	0.5000	0.1000	0.4000	0.2000	0.4000	0
	0.2500	0.7500	0	0	1.0000	0	0	0
	0.6667	0.1667	0.1667	0				
	1.0000	0	0	0				
val(:, :, 23) =					0.6154	0	0.2308	0.1538
					0.3333	0.6667	0	0
					1.0000	0	0	0
					1.0000	0	0	0
	0.5000	0	0.4000	0.1000				
	0.5000	0.5000	0	0				
	0.3750	0.1250	0.5000	0				
	1.0000	0	0	0				
val(:, :, 24) =					0.7500	0	0.1875	0.0625
					1.0000	0	0	0
					1.0000	0	0	0
					1.0000	0	0	0
	0.5455	0	0.3636	0.0909				
	0.5000	0.5000	0	0				
	0.4286	0.1429	0.4286	0				
	1.0000	0	0	0				
val(:, :, 25) =					0.5714	0	0.2857	0.1429
					0	0.9000	0.1000	0
					0.6667	0.3333	0	0
					1.0000	0	0	0
	0.5833	0	0.3333	0.0833				
	1.0000	0	0	0				
	0.5714	0	0.4286	0				
	1.0000	0	0	0				
val(:, :, 26) =					0.5714	0	0.4286	0
					0	0.9000	0.1000	0
					0.7500	0.2500	0	0
					0	0	0	0
	0.5455	0	0.4545	0				
	0.5000	0.5000	0	0				
	0.6250	0	0.3750	0				
	0	0	0	0				
val(:, :, 27) =					0.4286	0	0.5714	0
					0	0.8889	0.1111	0
					0.8000	0.2000	0	0
					0	0	0	0
	0.6154	0	0.3077	0.0769				
	0	0	0	0				
	0.5714	0	0.4286	0				
	1.0000	0	0	0				
val(:, :, 28) =					0.3750	0.1250	0.5000	0
					0	0.8750	0.1250	0
					1.0000	0	0	0
					0	0	0	0
	0.5455	0	0.3636	0.0909				
	0.5000	0.5000	0	0				
	0.4286	0.1429	0.4286	0				
	1.0000	0	0	0				
val(:, :, 29) =					0.4286	0	0.4286	0.1429
					0	0.8889	0.1111	0
					0.7500	0.2500	0	0
					1.0000	0	0	0
	0.5455	0	0.4545	0				
	0.5000	0.5000	0	0				
	0.6250	0	0.3750	0				
	0	0	0	0				
val(:, :, 30) =					0.4286	0	0.4286	0.1429
					0	0.8889	0.1111	0
					0.7500	0.2500	0	0
					1.0000	0	0	0
	0.6000	0	0.3000	0.1000				
	0.3333	0.3333	0.3333	0				
	0.2857	0.1429	0.5714	0				
	1.0000	0	0	0				
val(:, :, 31) =					0.3750	0	0.5000	0.1250
					0.1429	0.8571	0	0
					0.6000	0.2000	0.2000	0
					1.0000	0	0	0
	0.4286	0	0.4286	0.1429				
	0.1250	0.7500	0.1250	0				

0	0	1.0000	0	val(:, :, 69) =				
					0.4000	0	0.4000	0.2000
val(:, :, 60) =					0.1250	0.7500	0.1250	0
					0.4286	0.1429	0.4286	0
	0.5556	0	0.4444	0	0	0	1.0000	0
	0.2000	0.8000	0	0				
	0.5000	0.1667	0.3333	0				
	0	1.0000	0	0				
				val(:, :, 70) =				
					0.6250	0	0.2500	0.1250
					0	0.8333	0.1667	0
val(:, :, 61) =					0.5000	0	0.5000	0
	0.4286	0	0.4286	0.1429	1.0000	0	0	0
	0.1111	0.7778	0.1111	0				
	0.5000	0.5000	0	0				
	1.0000	0	0	0				
				val(:, :, 71) =				
					0.6250	0	0.2500	0.1250
					0	0.8333	0.1667	0
val(:, :, 62) =					0.5000	0	0.5000	0
	0.3750	0.1250	0.5000	0	1.0000	0	0	0
	0.2222	0.7778	0	0				
	0.7500	0.2500	0	0				
	0	0	0	0				
				val(:, :, 72) =				
					0.6667	0	0.2500	0.0833
					0.5000	0.5000	0	0
val(:, :, 63) =					0.5000	0.1667	0.3333	0
	0.4286	0	0.4286	0.1429	0	0	1.0000	0
	0.1000	0.9000	0	0				
	0.6667	0.3333	0	0				
	1.0000	0	0	0				
				val(:, :, 73) =				
					0.6667	0	0.2500	0.0833
					0.3333	0.6667	0	0
val(:, :, 64) =					0.6000	0.2000	0.2000	0
	0.4286	0	0.4286	0.1429	0	0	1.0000	0
	0.1000	0.9000	0	0				
	0.6667	0.3333	0	0				
	1.0000	0	0	0				
				val(:, :, 74) =				
					0.6667	0	0.2500	0.0833
					0.3333	0.6667	0	0
val(:, :, 65) =					0.6000	0.2000	0.2000	0
	0.4286	0	0.5714	0	0	0	1.0000	0
	0.1250	0.7500	0.1250	0				
	0.5000	0.3333	0.1667	0				
	0	0	0	0				
				val(:, :, 75) =				
					0.5000	0	0.3333	0.1667
					0.1429	0.8571	0	0
val(:, :, 66) =					0.1429	0.1429	0.7143	0
	0.4444	0	0.5556	0	1.0000	0	0	0
	0.1667	0.8333	0	0				
	0.8333	0	0.1667	0				
	0	0	0	0				
				val(:, :, 76) =				
					0.5000	0	0.3333	0.1667
					0	0.8750	0.1250	0
val(:, :, 67) =					0.3333	0.1667	0.5000	0
	0.5000	0.1250	0.3750	0	1.0000	0	0	0
	0.1111	0.8889	0	0				
	0.7500	0	0.2500	0				
	0	0	0	0				
				val(:, :, 77) =				
					0.5000	0	0.3333	0.1667
					0	0.8750	0.1250	0
val(:, :, 68) =					0.3333	0.1667	0.5000	0
	0.4286	0	0.5714	0	1.0000	0	0	0
	0.1111	0.8889	0	0				
	0.6000	0.2000	0.2000	0				
	0	0	0	0				
				val(:, :, 78) =				
					0.5000	0	0.3333	0.1667

0	0.8750	0.1250	0				
0.3333	0.1667	0.5000	0				
1.0000	0	0	0				
val(:, :, 79) =							
0.5000	0	0.3333	0.1667	0.6000	0	0.4000	0
0.1250	0.7500	0.1250	0	0.2500	0.7500	0	0
0.3333	0.1667	0.5000	0	0.5000	0	0.5000	0
1.0000	0	0	0	0	1.0000	0	0
val(:, :, 80) =							
0.5000	0	0.3333	0.1667	0.6000	0	0.3000	0.1000
0	0.8750	0.1250	0	0.2500	0.7500	0	0
0.3333	0.1667	0.5000	0	0.4000	0	0.6000	0
1.0000	0	0	0	0.5000	0.5000	0	0
val(:, :, 81) =							
0.4000	0.2000	0.2000	0.2000	0.6000	0	0.3000	0.1000
0	0.8000	0.2000	0	0.2500	0.7500	0	0
0.4000	0.2000	0.4000	0	0.4000	0	0.6000	0
1.0000	0	0	0	0.5000	0.5000	0	0
val(:, :, 82) =							
0.4000	0.2000	0.4000	0	0.5000	0	0.5000	0
0	0.8000	0.2000	0	0.2500	0.7500	0	0
0.5000	0.1667	0.3333	0	0.6667	0	0.3333	0
0	0	0	0	0	1.0000	0	0
val(:, :, 83) =							
0.4000	0.2000	0.2000	0.2000	0.6364	0.0909	0.1818	0.0909
0	0.7778	0.2222	0	0	0.6000	0.4000	0
0.3333	0.1667	0.5000	0	0.7500	0.2500	0	0
1.0000	0	0	0	1.0000	0	0	0
val(:, :, 84) =							
0.5000	0	0.3333	0.1667	0.6667	0.0833	0.2500	0
0	0.8750	0.1250	0	0	0.5000	0.5000	0
0.3333	0.1667	0.5000	0	0.8000	0.2000	0	0
1.0000	0	0	0	0	0	0	0
val(:, :, 85) =							
0.5000	0	0.3333	0.1667	0.6000	0.1000	0.3000	0
0	0.8750	0.1250	0	0.1667	0.5000	0.3333	0
0.3333	0.1667	0.5000	0	0.6000	0.4000	0	0
1.0000	0	0	0	0	0	0	0
val(:, :, 86) =							
0.3333	0	0.5000	0.1667	0.5000	0.1667	0.3333	0
0.1111	0.6667	0.2222	0	0.5000	0.2500	0.2500	0
0.4000	0.4000	0.2000	0	0.6000	0.4000	0	0
1.0000	0	0	0	0	0	0	0
val(:, :, 87) =							
0.6000	0	0.3000	0.1000	0.6364	0.0909	0.2727	0
0.2500	0.7500	0	0	0	0.6000	0.4000	0
0.4000	0	0.6000	0	0.6000	0.4000	0	0
0.5000	0.5000	0	0	0	0	0	0
val(:, :, 88) =							
0.6000	0	0.4000	0	0.6000	0	0.4000	0
0.2500	0.7500	0	0	0.2500	0.7500	0	0
0.5000	0	0.5000	0	0.5000	0	0.5000	0
0	1.0000	0	0	0	1.0000	0	0
val(:, :, 89) =							
0.6000	0	0.3000	0.1000	0.6000	0	0.3000	0.1000
0.2500	0.7500	0	0	0.2500	0.7500	0	0
0.4000	0	0.6000	0	0.4000	0	0.6000	0
0.5000	0.5000	0	0	0.5000	0.5000	0	0
val(:, :, 90) =							
0.6000	0	0.3000	0.1000	0.6000	0	0.3000	0.1000
0.2500	0.7500	0	0	0.2500	0.7500	0	0
0.4000	0	0.6000	0	0.4000	0	0.6000	0
0.5000	0.5000	0	0	0.5000	0.5000	0	0
val(:, :, 91) =							
0.5000	0	0.5000	0	0.6000	0	0.3000	0.1000
0.2500	0.7500	0	0	0.2500	0.7500	0	0
0.6667	0	0.3333	0	0.4000	0	0.6000	0
0	1.0000	0	0	0.5000	0.5000	0	0
val(:, :, 92) =							
0.6364	0.0909	0.1818	0.0909	0.6364	0.0909	0.1818	0.0909
0	0.6000	0.4000	0	0	0.6000	0.4000	0
0.7500	0.2500	0	0	0.7500	0.2500	0	0
1.0000	0	0	0	1.0000	0	0	0
val(:, :, 93) =							
0.6667	0.0833	0.2500	0	0.6667	0.0833	0.2500	0
0	0.5000	0.5000	0	0	0.5000	0.5000	0
0.8000	0.2000	0	0	0.8000	0.2000	0	0
0	0	0	0	0	0	0	0
val(:, :, 94) =							
0.6000	0.1000	0.3000	0	0.6000	0.1000	0.3000	0
0.1667	0.5000	0.3333	0	0.1667	0.5000	0.3333	0
0.6000	0.4000	0	0	0.6000	0.4000	0	0
0	0	0	0	0	0	0	0
val(:, :, 95) =							
0.5000	0.1667	0.3333	0	0.5000	0.1667	0.3333	0
0.5000	0.2500	0.2500	0	0.5000	0.2500	0.2500	0
0.6000	0.4000	0	0	0.6000	0.4000	0	0
0	0	0	0	0	0	0	0
val(:, :, 96) =							
0.6364	0.0909	0.2727	0	0.6364	0.0909	0.2727	0
0	0.6000	0.4000	0	0	0.6000	0.4000	0
0.6000	0.4000	0	0	0.6000	0.4000	0	0
0	0	0	0	0	0	0	0
val(:, :, 97) =							

0.5833	0	0.4167	0
0.5000	0.5000	0	0
0.7143	0	0.2857	0
0	0	0	0

1.0000	0	0	0
0	0	1.0000	0

T1011 – Praga 3

val(:, :, 98) =

0.5000	0.1250	0.2500	0.1250
0.1429	0.7143	0.1429	0
0.6000	0	0.4000	0
1.0000	0	0	0

val(:, :, 1) =

0.3333	0	0.6667	0
0.1000	0.8000	0.1000	0
0.6000	0.4000	0	0
0	0	0	0

val(:, :, 99) =

0.4286	0.1429	0.2857	0.1429
0.2500	0.6250	0.1250	0
0.4000	0.2000	0.4000	0
1.0000	0	0	0

val(:, :, 2) =

0.4444	0.1111	0.3333	0.1111
0.1429	0.7143	0.1429	0
0.7500	0.2500	0	0
0	1.0000	0	0

val(:, :, 100) =

0.5556	0	0.3333	0.1111
0.2500	0.7500	0	0
0.5000	0.1667	0.3333	0
0	0	0.5000	0.5000

val(:, :, 3) =

0.1667	0.1667	0.6667	0
0.1000	0.8000	0.1000	0
0.6000	0.4000	0	0
0	0	0	0

val(:, :, 101) =

0.7000	0	0.2000	0.1000
0	0.6667	0.3333	0
0.3333	0.3333	0.3333	0
0	0	0.5000	0.5000

val(:, :, 4) =

0.6667	0	0.3333	0
0	0.7500	0.1667	0.0833
0.4000	0.2000	0.4000	0
0	1.0000	0	0

val(:, :, 102) =

0.7500	0	0.1875	0.0625
0	0	0	0
1.0000	0	0	0
0	0	1.0000	0

val(:, :, 5) =

0.5000	0	0.3333	0.1667
0.1111	0.7778	0.1111	0
0.6000	0	0.4000	0
0	1.0000	0	0

val(:, :, 103) =

0.7500	0	0.1875	0.0625
0	0	0	0
1.0000	0	0	0
0	0	1.0000	0

val(:, :, 6) =

0.7647	0.0588	0.1765	0
1.0000	0	0	0
1.0000	0	0	0
0	0	0	0

val(:, :, 104) =

0.7500	0	0.1875	0.0625
0	0	0	0
1.0000	0	0	0
0	0	1.0000	0

val(:, :, 7) =

0	0	1.0000	0
0	0.6667	0.3333	0
0.1250	0.5000	0.3750	0
0	0	0	0

val(:, :, 105) =

0.7500	0	0.1875	0.0625
0	0	0	0
1.0000	0	0	0
0	0	1.0000	0

val(:, :, 8) =

0	0.5000	0.5000	0
0.1818	0.5455	0.1818	0.0909
0.2000	0.8000	0	0
0	0	1.0000	0

val(:, :, 106) =

0.7500	0	0.1875	0.0625
0	0	0	0

val(:, :, 9) =

0.4286	0.1429	0.2857	0.1429
0.1250	0.6250	0.2500	0
0.4000	0.6000	0	0


```

0      0      1.0000      0
val(:, :, 10) =
0.4000  0.4000  0.2000  0
0.1000  0.5000  0.4000  0
0.1667  0.6667  0.1667  0
0      0      0      0
val(:, :, 11) =
0.5000  0.1667  0.1667  0.1667
0      0.6667  0.3333  0
0.4000  0.6000  0      0
0      0      1.0000  0
val(:, :, 12) =
0.7500      0      0.2500  0
0      0.7500  0.2500  0
0.2222  0.1111  0.6667  0
0      0      0      0
val(:, :, 13) =
0.7778  0.1111      0      0.1111
0      0.6000  0.4000  0
0.3333  0.1667  0.5000  0
0      0      1.0000  0
val(:, :, 14) =
0.4286  0.2857  0.2857  0
0.1429  0.7143  0.1429  0
0.2857  0.1429  0.5714  0
0      0      0      0
val(:, :, 15) =
0.2500  0.2500  0.5000  0
0.1000  0.8000  0.1000  0
0.2857  0.1429  0.5714  0
0      0      0      0
val(:, :, 16) =
0.3333  0.3333  0.3333  0
0.0909  0.5455  0.3636  0
0      0.7143  0.2857  0
0      0      0      0
val(:, :, 17) =
0.3333  0.3333  0.3333  0
0.0833  0.5833  0.2500  0.0833
0      0.8000  0.2000  0
0      1.0000      0      0
val(:, :, 18) =
0.5000  0.1667  0.1667  0.1667
0.2222  0.4444  0.3333  0
0      0.8000  0.2000  0
0      1.0000      0      0
val(:, :, 19) =
0.6667  0.1667  0.1667  0
0      0.5556  0.3333  0.1111
0.2000  0.6000  0.2000  0
0      1.0000      0      0
val(:, :, 20) =
0.1667  0.3333  0.5000  0
0.1250  0.5000  0.2500  0.1250
0.5000  0.3333  0.1667  0
0      1.0000      0      0
val(:, :, 21) =
0.3750  0.1250  0.5000  0
0.1667  0.5000  0.3333  0
0.4286  0.4286  0.1429  0
0      0      0      0
val(:, :, 22) =
0.3333  0.1667  0.5000  0
0      0.6250  0.2500  0.1250
0.5000  0.3333  0.1667  0
0      1.0000      0      0
val(:, :, 23) =
0.3750  0.2500  0.2500  0.1250
0.5000      0      0.5000  0
0.2500  0.2500  0.5000  0
0      1.0000      0      0
val(:, :, 24) =
0.3750  0.2500  0.2500  0.1250
0.2000  0.4000  0.4000  0
0.4286  0.1429  0.4286  0
0      1.0000      0      0
val(:, :, 25) =
0.4286  0.2857  0.1429  0.1429
0.1667  0.3333  0.5000  0
0.2857  0.2857  0.4286  0
0      1.0000      0      0
val(:, :, 26) =
0.5556  0.1111  0.3333  0
0.2500  0.2500  0.5000  0
0.2500  0.3750  0.3750  0
0      0      0      0
val(:, :, 27) =
0.4444  0.2222  0.2222  0.1111
0.2500  0.2500  0.5000  0
0.5714      0      0.4286  0
0      1.0000      0      0
val(:, :, 28) =
0.6000  0.2000  0.1000  0.1000

```


0.4000	0	0.6000	0	0.8000	0	0.2000	0
0.1111	0.6667	0	0.2222	1.0000	0	0	0
0.7500	0.2500	0	0				
0	0.3333	0.3333	0.3333	val(:, :, 57) =			
val(:, :, 48) =				0.3333	0	0.6667	0
0.5000	0	0.5000	0	0.2222	0.3333	0.4444	0
0	0.7500	0.1250	0.1250	0	0.6667	0.3333	0
0.8000	0	0.2000	0	0	0	0	0
0	0.5000	0	0.5000	val(:, :, 58) =			
val(:, :, 49) =				0	0	1.0000	0
0.4000	0	0.6000	0	0.0909	0.4545	0.3636	0.0909
0.1111	0.6667	0.1111	0.1111	0	0.7500	0.2500	0
0.6000	0.2000	0.2000	0	0	0	1.0000	0
0	0.5000	0	0.5000	val(:, :, 59) =			
val(:, :, 50) =				0	0	1.0000	0
0.5714	0.1429	0.2857	0	0.0909	0.4545	0.3636	0.0909
0.2500	0.5000	0	0.2500	0	0.7500	0.2500	0
0.6667	0.3333	0	0	0	0	1.0000	0
0	0.3333	0.3333	0.3333	val(:, :, 60) =			
val(:, :, 51) =				0.7143	0	0.2857	0
0.7692	0.0769	0.1538	0	0	0	0	0
0	0.5000	0.5000	0	0.6667	0	0.3333	0
0.4000	0.2000	0.4000	0	1.0000	0	0	0
1.0000	0	0	0	val(:, :, 61) =			
val(:, :, 52) =				0.6000	0.1000	0.2000	0.1000
0.2500	0	0.7500	0	0	0.6667	0.3333	0
0.2222	0.4444	0.2222	0.1111	0.5000	0.5000	0	0
0.3333	0.5000	0	0.1667	1.0000	0	0	0
0	0.5000	0.5000	0	val(:, :, 62) =			
val(:, :, 53) =				0.5455	0.1818	0.2727	0
0.3333	0	0.6667	0	0	0.8333	0.1667	0
0.4286	0.2857	0.2857	0	1.0000	0	0	0
0.2857	0.4286	0.1429	0.1429	0	0	0	0
0	1.0000	0	0	val(:, :, 63) =			
val(:, :, 54) =				0.7000	0.1000	0.2000	0
0.3333	0	0.6667	0	0	0.7143	0.1429	0.1429
0.1111	0.4444	0.4444	0	0.3333	0.6667	0	0
0.2500	0.3750	0.2500	0.1250	1.0000	0	0	0
0	1.0000	0	0	val(:, :, 64) =			
val(:, :, 55) =				0.6364	0.1818	0.1818	0
1	0	0	0	0	0.6667	0.1667	0.1667
0	0	0	0	0.6667	0.3333	0	0
0	0	0	0	1.0000	0	0	0
1	0	0	0	val(:, :, 65) =			
val(:, :, 56) =				0.5000	0.1667	0.3333	0
0.6667	0.0667	0.2667	0	0.1111	0.5556	0.3333	0
0	0	0	0	0.1667	0.6667	0.1667	0
				0	0	0	0

1.0000	0	0	0	val(:, :, 94) =				
					0.6875	0	0.3125	0
val(:, :, 85) =					0	0	0	0
					1.0000	0	0	0
0.6154	0.0769	0.2308	0.0769		0	0	0	0
1.0000	0	0	0	val(:, :, 95) =				
0.5000	0	0.5000	0		0.6875	0	0.3125	0
1.0000	0	0	0		0	0	0	0
					1.0000	0	0	0
val(:, :, 86) =					0	0	0	0
0.6000	0	0.3333	0.0667	val(:, :, 96) =				
0	0	0	0		0.6875	0	0.3125	0
0.8000	0	0.2000	0		0	0	0	0
1.0000	0	0	0		1.0000	0	0	0
					0	0	0	0
val(:, :, 87) =				val(:, :, 97) =				
0.4444	0.2222	0.2222	0.1111		0.6875	0	0.3125	0
0.2000	0.6000	0.2000	0		0	0	0	0
0.4000	0	0.6000	0		1.0000	0	0	0
1.0000	0	0	0		0	0	0	0
				val(:, :, 98) =				
val(:, :, 88) =					0.6429	0	0.3571	0
0.5000	0.2500	0.2500	0		0	0	0	0
0	0.6667	0.3333	0		0.7143	0	0.2857	0
0.5000	0	0.5000	0		0	0	0	0
1.0000	0	0	0	val(:, :, 99) =				
					0.7333	0	0.2000	0.0667
val(:, :, 89) =					0	0	0	0
0.6923	0.0769	0.1538	0.0769		0.6000	0	0.4000	0
0	0	1.0000	0		1.0000	0	0	0
0.4000	0	0.6000	0	val(:, :, 100) =				
1.0000	0	0	0		0.7333	0	0.2000	0.0667
					0	0	0	0
val(:, :, 90) =					0.6000	0	0.4000	0
0.6923	0.0769	0.1538	0.0769		1.0000	0	0	0
0	0	1.0000	0	val(:, :, 101) =				
0.4000	0	0.6000	0		0.6667	0	0.2500	0.0833
1.0000	0	0	0		1.0000	0	0	0
					0.5000	0.1667	0.3333	0
val(:, :, 91) =					0	0	0.5000	0.5000
0.5000	0.1667	0.3333	0	val(:, :, 102) =				
0.5000	0	0.5000	0		0.6667	0	0.2500	0.0833
0.6667	0	0.3333	0		1.0000	0	0	0
1.0000	0	0	0		0.5000	0.1667	0.3333	0
					0	0	0.5000	0.5000
val(:, :, 92) =				val(:, :, 103) =				
0.6875	0	0.2500	0.0625		0.7500	0	0.1875	0.0625
0	0	0	0		0	0	0	0
1.0000	0	0	0		1.0000	0	0	0
1.0000	0	0	0		0	0	1.0000	0
val(:, :, 93) =					0.7500	0	0.1875	0.0625
0.6875	0	0.3125	0		0	0	0	0
0	0	0	0		1.0000	0	0	0
1.0000	0	0	0		0	0	1.0000	0
0	0	0	0					
					0.7500	0	0.1875	0.0625

0	0	0	0	0.7500	0	0	0.2500
1.0000	0	0	0	0	0.5000	0	0.5000
0	0	1.0000	0				
val(:, :, 104) =				val(:, :, 7) =			
0.7500	0	0.1875	0.0625	0.6429	0	0.3571	0
0	0	0	0	0.5000	0	0	0.5000
1.0000	0	0	0	0.8000	0.2000	0	0
0	0	1.0000	0	0	1.0000	0	0
val(:, :, 105) =				val(:, :, 8) =			
0.7500	0	0.1875	0.0625	0.6000	0	0.3333	0.0667
0	0	0	0	1.0000	0	0	0
1.0000	0	0	0	1.0000	0	0	0
0	0	1.0000	0	0	1.0000	0	0
val(:, :, 106) =				val(:, :, 9) =			
0.7500	0	0.1875	0.0625	0.6429	0	0.3571	0
0	0	0	0	0.5000	0	0	0.5000
1.0000	0	0	0	0.8000	0.2000	0	0
0	0	1.0000	0	0	1.0000	0	0
<u>T1112 – Praga 1</u>				val(:, :, 10) =			
val(:, :, 1) =				0.6429	0	0.3571	0
0.7500	0.0625	0.1875	0	0.5000	0	0	0.5000
0.5000	0	0	0.5000	0.8000	0.2000	0	0
1.0000	0	0	0	0	1.0000	0	0
0	1.0000	0	0	val(:, :, 11) =			
val(:, :, 2) =				0.6250	0	0.3125	0.0625
0.7500	0.0625	0.1875	0	0	0	0	0
0.5000	0	0	0.5000	1.0000	0	0	0
1.0000	0	0	0	1.0000	0	0	0
0	1.0000	0	0	val(:, :, 12) =			
val(:, :, 3) =				0.6429	0	0.3571	0
0.7500	0.0625	0.1875	0	0	0	0	0
0.5000	0	0	0.5000	0.5714	0	0.2857	0.1429
1.0000	0	0	0	1.0000	0	0	0
0	1.0000	0	0	val(:, :, 13) =			
val(:, :, 4) =				0.7143	0	0.2857	0
0.7333	0	0.2000	0.0667	1.0000	0	0	0
1.0000	0	0	0	0.6000	0	0.2000	0.2000
0.7500	0	0.2500	0	0	0.5000	0	0.5000
0	0.5000	0	0.5000	val(:, :, 14) =			
val(:, :, 5) =				0.7143	0	0.2857	0
0.7333	0	0.2000	0.0667	1.0000	0	0	0
1.0000	0	0	0	0.5000	0	0.3333	0.1667
0.7500	0	0.2500	0	0	1.0000	0	0
0	0.5000	0	0.5000	val(:, :, 15) =			
val(:, :, 6) =				0.6429	0	0.3571	0
0.6923	0	0.3077	0	1.0000	0	0	0
0.3333	0.6667	0	0	0.8000	0	0	0.2000
				0	0.5000	0	0.5000

0	1.0000	0	0	val(:, :, 44) =				
					0.9500	0	0.0500	0
val(:, :, 35) =					1.0000	0	0	0
					1.0000	0	0	0
0.6923	0	0.3077	0		0	0	0	0
0.3333	0.3333	0	0.3333					
0.6000	0.2000	0.2000	0	val(:, :, 45) =				
0	1.0000	0	0		0.7333	0	0.2667	0
					0	0	0	0
val(:, :, 36) =					0.5714	0	0.4286	0
					0	0	0	0
0.5333	0.0667	0.4000	0					
0	0	0	0	val(:, :, 46) =				
0.8571	0	0.1429	0		0.7333	0	0.2667	0
0	0	0	0		0	0	0	0
					0.5714	0	0.4286	0
val(:, :, 37) =					0	0	0	0
0.5333	0	0.4000	0.0667	val(:, :, 47) =				
0	0	0	0		0.7333	0	0.2667	0
0.8571	0	0.1429	0		0	0	0	0
0	0	0	0		0.5714	0	0.4286	0
					0	0	0	0
val(:, :, 38) =				val(:, :, 48) =				
					0.8333	0	0.1667	0
0.5333	0	0.4000	0.0667		0	0	0	0
0	0	0	0		0.6667	0	0	0.3333
0.8571	0	0.1429	0		1.0000	0	0	0
0	0	0	0	val(:, :, 49) =				
					0.8667	0	0.1333	0
val(:, :, 39) =					0	0.6667	0.3333	0
					0.6667	0	0	0.3333
0.5333	0	0.4000	0.0667		0	1.0000	0	0
0	0	0	0	val(:, :, 50) =				
0.8571	0	0.1429	0		0.7143	0.2143	0.0714	0
0	0	0	0		0.2500	0.2500	0.5000	0
					0.6667	0	0	0.3333
val(:, :, 40) =					1.0000	0	0	0
0.6667	0	0.2667	0.0667	val(:, :, 51) =				
0	0	0	0		0.6471	0.0588	0.2941	0
0.5714	0	0.4286	0		0	0	0	0
0	0	0	0		1.0000	0	0	0
					0	0	0	0
val(:, :, 41) =				val(:, :, 52) =				
					0.7333	0	0.2000	0.0667
0.6250	0	0.3125	0.0625		0	0	0	0
0	0	0	0		0.6667	0	0.3333	0
0.8333	0	0.1667	0		0	0	1.0000	0
0	0	0	0	val(:, :, 53) =				
					0.7333	0	0.2000	0.0667
val(:, :, 42) =					0	0	0	0
0.6250	0	0.3125	0.0625		0.6667	0	0.3333	0
0	0	0	0		0	0	1.0000	0
0.8333	0	0.1667	0					
0	0	0	0					
val(:, :, 43) =								
0.6471	0	0.2941	0.0588					
0	0	0	0					
1.0000	0	0	0					
0	0	0	0					


```

0      0      0      0
0.6667  0      0.3333  0
0      0      1.0000  0
val(:, :, 54) =
0.7333      0      0.2000  0.0667
1.0000      0      0      0
0.6000      0      0.4000  0
0      1.0000      0      0
val(:, :, 55) =
0.6250  0.0625  0.3125      0
0      0      0      0
0.8333      0      0.1667  0
0      0      0      0
val(:, :, 56) =
0.6000  0.0667  0.3333      0
0      0      0      0
0.7143      0      0.2857  0
0      0      0      0
val(:, :, 57) =
0.6667      0      0.3333      0
0      0      0      0
0.7143      0      0.2857  0
0      0      0      0
val(:, :, 58) =
0.6667      0      0.2667  0.0667
0      0      0      0
0.8333      0      0.1667  0
0      0      1.0000  0
val(:, :, 59) =
0.6667      0      0.3333      0
0      0      0      0
0.7143      0      0.2857  0
0      0      0      0
val(:, :, 60) =
0.7222  0.0556  0.2222      0
0      0      0      0
1.0000      0      0      0
0      0      0      0
val(:, :, 61) =
0.6429      0      0.3571      0
0      0      0      0
0.5714      0      0.2857  0.1429
1.0000      0      0      0
val(:, :, 62) =
0.6154      0      0.3846      0
1.0000      0      0      0
0.5714      0      0.2857  0.1429
0      1.0000      0      0
val(:, :, 63) =
0.6154      0      0.3846      0
1.0000      0      0      0
0.5714      0      0.2857  0.1429
0      1.0000      0      0
val(:, :, 64) =
0.5714      0      0.3571  0.0714
1.0000      0      0      0
0.8333      0      0.1667  0
0      1.0000      0      0
val(:, :, 65) =
0.6429      0      0.2857  0.0714
1.0000      0      0      0
0.6667      0      0.3333      0
0      1.0000      0      0
val(:, :, 66) =
0.6429      0      0.2857  0.0714
1.0000      0      0      0
0.6667      0      0.3333      0
0      1.0000      0      0
val(:, :, 67) =
0.6429      0      0.2857  0.0714
1.0000      0      0      0
0.6667      0      0.3333      0
0      1.0000      0      0
val(:, :, 68) =
0.6429      0      0.2857  0.0714
1.0000      0      0      0
0.6667      0      0.3333      0
0      1.0000      0      0
val(:, :, 69) =
0.5455      0      0.4545      0
0.3333  0.3333      0      0.3333
0.5714  0.1429  0.2857      0
0      1.0000      0      0
val(:, :, 70) =
0.5455      0      0.4545      0
0.3333  0.3333      0      0.3333
0.5714  0.1429  0.2857      0
0      1.0000      0      0
val(:, :, 71) =
0.5455      0      0.4545      0
0.3333  0.3333      0      0.3333
0.5714  0.1429  0.2857      0
0      1.0000      0      0
val(:, :, 72) =
0.5455      0      0.4545      0
0.3333  0.3333      0      0.3333
0.5714  0.1429  0.2857      0
0      1.0000      0      0

```

				1.0000	0	0	0
0.6154	0.0769	0.2308	0.0769	0	1.0000	0	0
0	0	0.5000	0.5000				
0.8000	0	0.2000	0				
0	1.0000	0	0				
				val(:, :, 82) =			
				0.7333	0	0.2000	0.0667
				0.5000	0.5000	0	0
				0.7500	0	0.2500	0
				0	1.0000	0	0
val(:, :, 73) =							
0.6154	0.0769	0.2308	0.0769				
0	0	0.5000	0.5000				
0.8000	0	0.2000	0				
0	1.0000	0	0				
				val(:, :, 83) =			
				0.7333	0	0.2000	0.0667
				0.5000	0.5000	0	0
				0.7500	0	0.2500	0
				0	1.0000	0	0
val(:, :, 74) =							
0.6429	0	0.2143	0.1429				
0	0	1.0000	0				
0.8000	0	0.2000	0				
0	1.0000	0	0				
				val(:, :, 84) =			
				0.7333	0	0.2000	0.0667
				0.5000	0.5000	0	0
				0.7500	0	0.2500	0
				0	1.0000	0	0
val(:, :, 75) =							
0.7143	0	0.2857	0				
0.5000	0.5000	0	0				
0.6000	0	0.2000	0.2000				
0	1.0000	0	0				
				val(:, :, 85) =			
				0.7333	0	0.2000	0.0667
				0.5000	0.5000	0	0
				0.7500	0	0.2500	0
				0	1.0000	0	0
val(:, :, 76) =							
0.7143	0	0.2857	0				
0.5000	0.5000	0	0				
0.6000	0	0.2000	0.2000				
0	1.0000	0	0				
				val(:, :, 86) =			
				0.6923	0	0.2308	0.0769
				0.3333	0.6667	0	0
				0.7500	0	0.2500	0
				0	0.5000	0	0.5000
val(:, :, 77) =							
0.7143	0	0.2857	0				
0.5000	0.5000	0	0				
0.6000	0	0.2000	0.2000				
0	1.0000	0	0				
				val(:, :, 87) =			
				0.7143	0	0.2857	0
				1.0000	0	0	0
				0.5000	0	0.3333	0.1667
				0	1.0000	0	0
val(:, :, 78) =							
0.7143	0	0.2143	0.0714				
0.3333	0.6667	0	0				
0.7500	0	0.2500	0				
0	1.0000	0	0				
				val(:, :, 88) =			
				0.6667	0.0667	0.2667	0
				0	0	0	0
				0.5714	0	0.4286	0
				0	0	0	0
val(:, :, 79) =							
0.7143	0	0.2857	0				
0.5000	0.5000	0	0				
0.6000	0	0.2000	0.2000				
0	1.0000	0	0				
				val(:, :, 89) =			
				0.6471	0.0588	0.2941	0
				0	0	0	0
				1.0000	0	0	0
				0	0	0	0
val(:, :, 80) =							
0.7143	0	0.2143	0.0714				
0.3333	0.6667	0	0				
0.7500	0	0.2500	0				
0	1.0000	0	0				
				val(:, :, 90) =			
				0.6250	0.0625	0.2500	0.0625
				0	0	0	0
				1.0000	0	0	0
				0	0	1.0000	0
val(:, :, 81) =							
0.6429	0	0.2857	0.0714				
0.3333	0.6667	0	0				

```

                                0.6923      0      0.3077      0
val(:, :, 91) =                0.5000      0      0      0.5000
                                0.5000      0.1667      0.3333      0
                                0      1.0000      0      0
    0.6250    0.0625    0.2500    0.0625
      0        0        0        0
    1.0000      0      0      0
      0        0    1.0000      0

                                0.6429      0      0.2857      0.0714
val(:, :, 92) =                1.0000      0      0      0
                                0.6667      0      0.3333      0
                                0      1.0000      0      0
    0.6154    0.0769    0.3077      0
      0      0.6667      0    0.3333
    0.8000      0    0.2000      0
      0    1.0000      0      0

                                0.4167      0      0.5833      0
val(:, :, 93) =                0      0      0      0
                                0.6000      0      0.4000      0
                                0      0      0      0
    0.5833    0.0833    0.3333      0
      0      0.6667      0    0.3333
    0.6667      0    0.3333      0
      0    1.0000      0      0

                                0.4167      0      0.5000      0.0833
val(:, :, 94) =                0      0      0      0
                                0.6667      0      0.3333      0
                                0      0      1.0000      0
    0.6923      0      0.3077      0
      0    1.0000      0      0
    0.5000      0      0.3333    0.1667
      0    0.5000      0      0.5000

                                0.5385      0      0.3846      0.0769
val(:, :, 95) =                0      0      0      0
                                0.6250      0      0.3750      0
                                0      0      0      1.0000
    0.6154      0      0.3846      0
      0    0.5000      0      0.5000
    0.6000    0.2000      0      0.2000
    0.5000    0.5000      0      0

                                0.5000      0      0.4167      0.0833
val(:, :, 96) =                0      0      0      0
                                0.5556      0      0.4444      0
                                0      0      1.0000      0
    0.6154      0      0.3846      0
      0    0.5000      0      0.5000
    0.6000    0.2000      0      0.2000
    0.5000    0.5000      0      0

                                0.5385      0      0.4615      0
val(:, :, 97) =                0      0      0      0
                                0.5556      0      0.4444      0
                                0      0      0      0
    0.5833      0      0.1667    0.2500
      0      0      0      1.0000
    0.5000      0      0.5000      0
    0.4000    0.4000      0      0.2000

                                0.7143    0.0714    0.2143      0
val(:, :, 98) =                0.5000    0.2500      0      0.2500
                                1.0000      0      0      0
                                0      1.0000      0      0
    0.5833      0      0.1667    0.2500
      0      0      0      1.0000
    0.5000      0      0.5000      0
    0.4000    0.4000      0      0.2000

                                0.7500    0.0625    0.1875      0
val(:, :, 99) =                0.5000      0      0      0.5000
                                1.0000      0      0      0
                                0      1.0000      0      0
    0.5833      0      0.1667    0.2500
      0      0      0      1.0000
    0.5000      0      0.5000      0
    0.4000    0.4000      0      0.2000

                                0.7500    0.0625    0.1875      0
val(:, :, 100) =               0.5000      0      0      0.5000

```

T1112 – Praga 2

val(:, :, 1) =

```

    0.7143    0.0714    0.2143      0
    0.5000    0.2500      0      0.2500
    1.0000      0      0      0
      0      1.0000      0      0

```

val(:, :, 2) =

```

    0.7500    0.0625    0.1875      0
    0.5000      0      0      0.5000
    1.0000      0      0      0
      0      1.0000      0      0

```

val(:, :, 3) =

```

    0.7500    0.0625    0.1875      0
    0.5000      0      0      0.5000

```

```

1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 4) =
0.7333      0      0.2000  0.0667
1.0000      0      0      0
0.7500      0      0.2500      0
      0  0.5000      0      0.5000
val(:, :, 5) =
0.7333      0      0.2000  0.0667
1.0000      0      0      0
0.7500      0      0.2500      0
      0  0.5000      0      0.5000
val(:, :, 6) =
0.6667      0      0.3333      0
0.2500  0.7500      0      0
0.7500      0      0      0.2500
      0  0.5000      0      0.5000
val(:, :, 7) =
0.4167  0.0833  0.4167  0.0833
0.2500  0.7500      0      0
1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 8) =
0.5000      0      0.4167  0.0833
0.2500  0.7500      0      0
1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 9) =
0.4000      0      0.5000  0.1000
0.1667  0.8333      0      0
1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 10) =
0.4000      0      0.5000  0.1000
0.3333  0.6667      0      0
0.8000  0.2000      0      0
      0  1.0000      0      0
val(:, :, 11) =
0.4545      0      0.4545  0.0909
0.2000  0.8000      0      0
1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 12) =
0.6154      0      0.3846      0
1.0000      0      0      0
0.5714      0      0.2857  0.1429
      0  1.0000      0      0
val(:, :, 13) =
0.7143      0      0.2857      0
1.0000      0      0      0
0.6000      0      0.2000  0.2000
      0  0.5000      0      0.5000
val(:, :, 14) =
0.7143      0      0.2857      0
1.0000      0      0      0
0.5000      0      0.3333  0.1667
      0  1.0000      0      0
val(:, :, 15) =
0.5833      0      0.4167      0
0.3333  0.6667      0      0
0.8000      0      0      0.2000
      0  0.5000      0      0.5000
val(:, :, 16) =
0.6667  0.0833  0.2500      0
0.2500  0.5000      0      0.2500
0.6000      0      0.4000      0
      0  1.0000      0      0
val(:, :, 17) =
0.6667  0.0833  0.2500      0
0.2500  0.5000      0      0.2500
0.6000      0      0.4000      0
      0  1.0000      0      0
val(:, :, 18) =
0.5833  0.0833  0.3333      0
0.2000  0.6000      0      0.2000
1.0000      0      0      0
      0  1.0000      0      0
val(:, :, 19) =
0.6923      0      0.3077      0
      0  0.6667      0      0.3333
0.6000  0.2000  0.2000      0
      0  1.0000      0      0
val(:, :, 20) =
0.6364      0      0.3636      0
0.2000  0.6000      0      0.2000
0.6000  0.2000  0.2000      0
      0  1.0000      0      0
val(:, :, 21) =
0.6923      0      0.3077      0
      0  0.6667      0      0.3333
0.6000  0.2000  0.2000      0
      0  1.0000      0      0
val(:, :, 22) =
0.6923      0      0.3077      0
      0  0.6667      0      0.3333
0.6000  0.2000  0.2000      0
      0  1.0000      0      0

```

0.6923	0	0.3077	0	0	1.0000	0	0
0	0.6667	0	0.3333				
0.6000	0.2000	0.2000	0				
0	1.0000	0	0				
val(:, :, 23) =				val(:, :, 32) =			
0.5833	0	0.3333	0.0833	0.6364	0	0.2727	0.0909
0.2000	0.8000	0	0	0.1667	0.8333	0	0
1.0000	0	0	0	0.7500	0.2500	0	0
0	1.0000	0	0	0	0	1.0000	0
val(:, :, 24) =				val(:, :, 33) =			
0.5833	0	0.3333	0.0833	0.6667	0	0.3333	0
0.2000	0.8000	0	0	0.2000	0.8000	0	0
1.0000	0	0	0	0.6000	0.2000	0.2000	0
0	1.0000	0	0	0	0	0	0
val(:, :, 25) =				val(:, :, 34) =			
0.5000	0.1000	0.3000	0.1000	0.5000	0	0.5000	0
0.1429	0.7143	0.1429	0	0	0.8000	0.1000	0.1000
1.0000	0	0	0	0.6000	0.2000	0.2000	0
0	1.0000	0	0	0	1.0000	0	0
val(:, :, 26) =				val(:, :, 35) =			
0.5000	0.1000	0.3000	0.1000	0.4286	0	0.4286	0.1429
0.1429	0.7143	0.1429	0	0	0.8889	0.1111	0
1.0000	0	0	0	0.8000	0	0.2000	0
0	1.0000	0	0	0	1.0000	0	0
val(:, :, 27) =				val(:, :, 36) =			
0.3333	0.1111	0.4444	0.1111	0.5333	0.0667	0.4000	0
0.2500	0.7500	0	0	0	0	0	0
1.0000	0	0	0	0.8571	0	0.1429	0
0	1.0000	0	0	0	0	0	0
val(:, :, 28) =				val(:, :, 37) =			
0.3750	0.1250	0.3750	0.1250	0.4615	0.0769	0.3846	0.0769
0.1429	0.7143	0.1429	0	0.5000	0	0.5000	0
0.6667	0	0.3333	0	0.7143	0.1429	0.1429	0
0	1.0000	0	0	0	0	0	0
val(:, :, 29) =				val(:, :, 38) =			
0.2857	0.1429	0.4286	0.1429	0.5333	0	0.4000	0.0667
0.1250	0.7500	0.1250	0	0	0	0	0
0.6667	0	0.3333	0	0.8571	0	0.1429	0
0	1.0000	0	0	0	0	0	0
val(:, :, 30) =				val(:, :, 39) =			
0.6364	0.0909	0.2727	0	0.3636	0.1818	0.3636	0.0909
0.2500	0.5000	0	0.2500	0.2500	0.2500	0.5000	0
0.5000	0	0.5000	0	0.7143	0.1429	0.1429	0
0	1.0000	0	0	0	0	0	0
val(:, :, 31) =				val(:, :, 40) =			
0.6667	0.0833	0.2500	0	0.6923	0.0769	0.1538	0.0769
0	0.6667	0	0.3333	0	0	1.0000	0
0.5000	0	0.5000	0	0.4286	0.1429	0.4286	0
				0	0	0	0

```

val(:,:,41) =
    0.5833    0.0833    0.2500    0.0833
         0    0.5000    0.5000         0
    0.6667    0.1667    0.1667         0
         0         0         0         0

val(:,:,42) =
    0.6250         0    0.3125    0.0625
         0         0         0         0
    0.8333         0    0.1667         0
         0         0         0         0

val(:,:,43) =
    0.6250    0.0625    0.2500    0.0625
         0         0    1.0000         0
    1.0000         0         0         0
         0         0         0         0

val(:,:,44) =
    0.9524         0    0.0476         0
         0         0         0         0
    1.0000         0         0         0
         0         0         0         0

val(:,:,45) =
    0.4286         0    0.5714         0
    0.1250    0.8750         0         0
    0.4286    0.1429    0.4286         0
         0         0         0         0

val(:,:,46) =
    0.4286         0    0.5714         0
    0.1250    0.8750         0         0
    0.4286    0.1429    0.4286         0
         0         0         0         0

val(:,:,47) =
    0.6364         0    0.3636         0
    0.2500    0.7500         0         0
    0.4286    0.1429    0.4286         0
         0         0         0         0

val(:,:,48) =
    0.7273    0.0909    0.1818         0
    0.1429    0.7143    0.1429         0
    0.6667         0         0    0.3333
         0    1.0000         0         0

val(:,:,49) =
    0.7273    0.0909    0.1818         0
    0.1429    0.7143    0.1429         0
    0.6667         0         0    0.3333
         0    1.0000         0         0

val(:,:,50) =
    0.7778    0.1111    0.1111         0

val(:,:,51) =
    0.6667         0    0.3333         0
         0    1.0000         0         0
    0.8000    0.2000         0         0
         0         0         0         0

val(:,:,52) =
    0.6667         0    0.2500    0.0833
         0    1.0000         0         0
    0.5000    0.1667    0.3333         0
         0         0    1.0000         0

val(:,:,53) =
    0.6364         0    0.2727    0.0909
    0.2500    0.7500         0         0
    0.5000    0.1667    0.3333         0
         0         0    1.0000         0

val(:,:,54) =
    0.6923         0    0.2308    0.0769
         0    1.0000         0         0
    0.4000    0.2000    0.4000         0
    1.0000         0         0         0

val(:,:,55) =
    0.6667         0    0.3333         0
         0    1.0000         0         0
    0.6667    0.1667    0.1667         0
         0         0         0         0

val(:,:,56) =
    0.6154    0.0769    0.3077         0
         0    0.5000    0.5000         0
    0.5714    0.1429    0.2857         0
         0         0         0         0

val(:,:,57) =
    0.6154         0    0.3846         0
         0    1.0000         0         0
    0.5714    0.1429    0.2857         0
         0         0         0         0

val(:,:,58) =
    0.6154         0    0.3077    0.0769
         0    1.0000         0         0
    0.6667    0.1667    0.1667         0
         0         0    1.0000         0

val(:,:,59) =
    0.6154         0    0.3846         0
         0    1.0000         0         0
    0.5714    0.1429    0.2857         0
         0         0         0         0

```

```

                                0.4000    0.1000    0.5000    0
                                0.2500    0.5000    0          0.2500
                                0.7143    0          0.2857    0
                                0          1.0000    0          0
val(:, :, 60) =
    0.7500    0    0.2500    0
    0          1.0000    0    0
    0.7500    0.2500    0    0
    0          0          0    0

                                0.5000    0    0.4167    0.0833
                                0          1.0000    0          0
                                0.7143    0    0.2857    0
                                0          1.0000    0          0
val(:, :, 70) =
    0.5000    0    0.4167    0.0833
    0          1.0000    0          0
    0.7143    0    0.2857    0
    0          1.0000    0          0

                                0.4545    0.0909    0.4545    0
                                0          0.6667    0          0.3333
                                0.7143    0    0.2857    0
                                0          1.0000    0          0
val(:, :, 71) =
    0.4545    0.0909    0.4545    0
    0          0.6667    0          0.3333
    0.7143    0    0.2857    0
    0          1.0000    0          0

                                0.5714    0.0714    0.2857    0.0714
                                0          0          0          1.0000
                                0.8000    0    0.2000    0
                                0.5000    0.5000    0          0
val(:, :, 72) =
    0.5714    0.0714    0.2857    0.0714
    0          0          0          1.0000
    0.8000    0    0.2000    0
    0.5000    0.5000    0          0

                                0.5714    0.0714    0.2857    0.0714
                                0          0          0          1.0000
                                0.8000    0    0.2000    0
                                0.5000    0.5000    0          0
val(:, :, 73) =
    0.5714    0.0714    0.2857    0.0714
    0          0          0          1.0000
    0.8000    0    0.2000    0
    0.5000    0.5000    0          0

                                0.5714    0.0714    0.2857    0.0714
                                0          0          0          1.0000
                                0.8000    0    0.2000    0
                                0.5000    0.5000    0          0
val(:, :, 74) =
    0.5714    0.0714    0.2857    0.0714
    0          0          0          1.0000
    0.8000    0    0.2000    0
    0.5000    0.5000    0          0

                                0.5714    0.0714    0.2857    0.0714
                                0          0          0          1.0000
                                0.8000    0    0.2000    0
                                0.5000    0.5000    0          0
val(:, :, 75) =
    0.5714    0.0714    0.2857    0.0714
    0          0          0          1.0000
    0.8000    0    0.2000    0
    0.5000    0.5000    0          0

                                0.6667    0    0.3333    0
                                0.2500    0.7500    0          0
                                0.6000    0    0.2000    0.2000
                                0          1.0000    0          0
val(:, :, 76) =
    0.6667    0    0.3333    0
    0.2500    0.7500    0          0
    0.6000    0    0.2000    0.2000
    0          1.0000    0          0

                                0.5556    0    0.4444    0
                                0.1429    0.8571    0          0
                                0.6000    0    0.2000    0.2000
                                0          1.0000    0          0
val(:, :, 77) =
    0.5556    0    0.4444    0
    0.1429    0.8571    0          0
    0.6000    0    0.2000    0.2000
    0          1.0000    0          0

                                0.6000    0    0.4000    0
                                0.1667    0.8333    0          0
                                0.6000    0    0.2000    0.2000
                                0          1.0000    0          0
val(:, :, 78) =
    0.6000    0    0.4000    0
    0.1667    0.8333    0          0
    0.6000    0    0.2000    0.2000
    0          1.0000    0          0

                                0.6364    0    0.2727    0.0909
                                0.1667    0.8333    0          0
val(:, :, 69) =
    0.6364    0    0.2727    0.0909
    0.1667    0.8333    0          0

```

```

0.7500      0      0.2500      0
   0      1.0000      0      0
val(:, :, 79) =
0.6000      0      0.4000      0
0.1667      0.8333      0      0
0.6000      0      0.2000      0.2000
   0      1.0000      0      0
val(:, :, 80) =
0.6000      0      0.3000      0.1000
0.1429      0.8571      0      0
0.7500      0      0.2500      0
   0      1.0000      0      0
val(:, :, 81) =
0.5000      0      0.4000      0.1000
0.1429      0.8571      0      0
1.0000      0      0      0
   0      1.0000      0      0
val(:, :, 82) =
0.6364      0      0.2727      0.0909
0.1667      0.8333      0      0
0.7500      0      0.2500      0
   0      1.0000      0      0
val(:, :, 83) =
0.6667      0      0.2500      0.0833
0.2000      0.8000      0      0
0.7500      0      0.2500      0
   0      1.0000      0      0
val(:, :, 84) =
0.6364      0      0.2727      0.0909
0.1667      0.8333      0      0
0.7500      0      0.2500      0
   0      1.0000      0      0
val(:, :, 85) =
0.6000      0      0.3000      0.1000
0.1667      0.8333      0      0
0.7500      0      0.2500      0
   0      0.5000      0      0.5000
val(:, :, 86) =
0.6000      0      0.4000      0
0.2000      0.8000      0      0
0.5000      0      0.3333      0.1667
   0      1.0000      0      0
val(:, :, 87) =
0.6000      0      0.4000      0
0.2000      0.8000      0      0
0.4286      0.1429      0.4286      0
   0      0      0      0
val(:, :, 88) =
0.5000      0      0.5000      0
0.1667      0.8333      0      0
0.6667      0.1667      0.1667      0
   0      0      0      0
val(:, :, 89) =
0.5455      0      0.4545      0
0.3333      0.6667      0      0
0.8000      0.2000      0      0
   0      0      0      0
val(:, :, 90) =
0.5000      0      0.4000      0.1000
0.3333      0.6667      0      0
0.8000      0.2000      0      0
   0      0      1.0000      0
val(:, :, 91) =
0.5000      0      0.4000      0.1000
0.1667      0.8333      0      0
0.8000      0.2000      0      0
   0      0      1.0000      0
val(:, :, 92) =
0.6429      0      0.2857      0.0714
0.5000      0.5000      0      0
0.8000      0      0.2000      0
1.0000      0      0      0
val(:, :, 93) =
0.6154      0      0.3077      0.0769
0.5000      0.5000      0      0
0.6667      0      0.3333      0
1.0000      0      0      0
val(:, :, 94) =
0.6364      0      0.3636      0
0.3333      0.6667      0      0
0.5000      0      0.3333      0.1667
0.5000      0      0      0.5000
val(:, :, 95) =
0.5455      0      0.4545      0
0.2500      0.5000      0      0.2500
0.6000      0.2000      0      0.2000
1.0000      0      0      0
val(:, :, 96) =
0.5000      0      0.5000      0
0.2000      0.6000      0      0.2000
0.6000      0.2000      0      0.2000
1.0000      0      0      0
val(:, :, 97) =

```


0.3750	0	0.1250	0.5000	0	0	0	0
0.2000	0.6000	0.2000	0				
0.2500	0.2500	0.5000	0				
0.6000	0.2000	0	0.2000				
val(:, :, 98) =							
0.2857	0	0.1429	0.5714	0.6154	0.0769	0.2308	0.0769
0.1667	0.6667	0.1667	0	0.2000	0.8000	0	0
0.2500	0.2500	0.5000	0	1.0000	0	0	0
0.6000	0.2000	0	0.2000	1.0000	0	0	0
val(:, :, 99) =							
0.2857	0	0.1429	0.5714	0.7647	0	0.1765	0.0588
0.1667	0.6667	0.1667	0	1.0000	0	0	0
0.2500	0.2500	0.5000	0	1.0000	0	0	0
0.6000	0.2000	0	0.2000	1.0000	0	0	0
val(:, :, 100) =							
0.5833	0	0.3333	0.0833	0.6667	0.0667	0.2000	0.0667
0.3333	0.6667	0	0	0.3333	0.6667	0	0
0.6667	0	0.3333	0	1.0000	0	0	0
0	1.0000	0	0	1.0000	0	0	0
val(:, :, 101) =							
0.5833	0	0.3333	0.0833	0.6667	0.0667	0.2000	0.0667
0.3333	0.6667	0	0	0	1.0000	0	0
0.6667	0	0.3333	0	0.7500	0	0.2500	0
0	1.0000	0	0	0.5000	0	0	0.5000
val(:, :, 102) =							
0.4444	0	0.5556	0	0.6667	0.0667	0.2000	0.0667
0	0.3333	0.6667	0	1.0000	0	0	0
0.4000	0.2000	0.4000	0	0.7500	0	0.2500	0
0	0	0	0	0.5000	0	0	0.5000
val(:, :, 103) =							
0.4444	0	0.5556	0	0.7500	0	0.2500	0
0	0.3333	0.3333	0.3333	0	0	0	0
0.4444	0.2222	0.3333	0	0.7500	0	0	0.2500
0	0	1.0000	0	0.5000	0	0	0.5000
val(:, :, 104) =							
0.5556	0	0.3333	0.1111	0.4667	0.1333	0.3333	0.0667
0	0.5000	0.5000	0	1.0000	0	0	0
0.3750	0.2500	0.3750	0	1.0000	0	0	0
0	0	0	1.0000	1.0000	0	0	0
val(:, :, 105) =							
0.5000	0	0.4000	0.1000	0.5000	0	0.4167	0.0833
0	0.5000	0.5000	0	0.2500	0.7500	0	0
0.4444	0.1111	0.4444	0	1.0000	0	0	0
0	0	1.0000	0	1.0000	0	0	0
val(:, :, 106) =							
0.5000	0	0.5000	0	0.5385	0	0.3846	0.0769
0	0.6000	0.4000	0	0.3333	0.6667	0	0
0.3333	0.2222	0.4444	0	1.0000	0	0	0
				1.0000	0	0	0

T1112 - Praga 3

val(:, :, 1) =

0.6154	0.0769	0.2308	0.0769
0.2000	0.8000	0	0
1.0000	0	0	0
1.0000	0	0	0

val(:, :, 2) =

0.7647	0	0.1765	0.0588
1.0000	0	0	0
1.0000	0	0	0
1.0000	0	0	0

val(:, :, 3) =

0.6667	0.0667	0.2000	0.0667
0.3333	0.6667	0	0
1.0000	0	0	0
1.0000	0	0	0

val(:, :, 4) =

0.6667	0.0667	0.2000	0.0667
0	1.0000	0	0
0.7500	0	0.2500	0
0.5000	0	0	0.5000

val(:, :, 5) =

0.6667	0.0667	0.2000	0.0667
1.0000	0	0	0
0.7500	0	0.2500	0
0.5000	0	0	0.5000

val(:, :, 6) =

0.7500	0	0.2500	0
0	0	0	0
0.7500	0	0	0.2500
0.5000	0	0	0.5000

val(:, :, 7) =

0.4667	0.1333	0.3333	0.0667
1.0000	0	0	0
1.0000	0	0	0
1.0000	0	0	0

val(:, :, 8) =

0.5000	0	0.4167	0.0833
0.2500	0.7500	0	0
1.0000	0	0	0
1.0000	0	0	0

val(:, :, 9) =

0.5385	0	0.3846	0.0769
0.3333	0.6667	0	0
1.0000	0	0	0
1.0000	0	0	0

1.0000	0	0	0	val(:, :, 38) =	0.4000	0.1000	0.4000	0.1000	
val(:, :, 29) =	0.6923	0	0.2308	0.0769	0	0.6000	0.4000	0	
	0	0.5000	0.5000	0	0.7143	0.1429	0.1429	0	
	0.6667	0	0.3333	0	0	0	0	0	
	1.0000	0	0	0	val(:, :, 39) =	0.4000	0.1000	0.4000	0.1000
val(:, :, 30) =	0.6667	0	0.2500	0.0833	0	0.6000	0.4000	0	
	0.3333	0.6667	0	0	0.7143	0.1429	0.1429	0	
	0.5000	0	0.5000	0	0	0	0	0	
	1.0000	0	0	0	val(:, :, 40) =	0.5000	0.1250	0.2500	0.1250
val(:, :, 31) =	0.6364	0	0.2727	0.0909	0	0.7143	0.2857	0	
	0.2500	0.7500	0	0	0.4286	0.1429	0.4286	0	
	0.5000	0	0.5000	0	0	0	0	0	
	1.0000	0	0	0	val(:, :, 41) =	0.4444	0.1111	0.3333	0.1111
val(:, :, 32) =	0.6364	0	0.2727	0.0909	0	0.7143	0.2857	0	
	0.3333	0.6667	0	0	0.6667	0.1667	0.1667	0	
	0.7500	0.2500	0	0	0	0	0	0	
	0	0	1.0000	0	val(:, :, 42) =	0.3636	0.0909	0.4545	0.0909
val(:, :, 33) =	0.7143	0	0.2857	0	0.2000	0.8000	0	0	
	0.3333	0.6667	0	0	0.8333	0	0.1667	0	
	0.8000	0	0.2000	0	0	0	0	0	
	0	0	0	0	val(:, :, 43) =	0.4545	0.0909	0.3636	0.0909
val(:, :, 34) =	0.5714	0	0.2857	0.1429	0	0.8333	0.1667	0	
	0.1111	0.6667	0.2222	0	1.0000	0	0	0	
	0.4000	0.4000	0.2000	0	0	0	0	0	
	1.0000	0	0	0	val(:, :, 44) =	0.8750	0.0625	0.0625	0
val(:, :, 35) =	0.6000	0	0.3000	0.1000	0.6000	0.4000	0	0	
	0	0.8333	0.1667	0	0	1.0000	0	0	
	0.8000	0	0.2000	0	0	0	0	0	
	1.0000	0	0	0	val(:, :, 45) =	0.7143	0	0.2857	0
val(:, :, 36) =	0.5000	0.1250	0.3750	0	1.0000	0	0	0	
	0	0.5714	0.4286	0	0.5714	0	0.4286	0	
	0.5714	0.2857	0.1429	0	0	0	0	0	
	0	0	0	0	val(:, :, 46) =	0.6154	0.0769	0.3077	0
val(:, :, 37) =	0.3636	0.0909	0.4545	0.0909	1.0000	0	0	0	
	0	0.7500	0.2500	0	0.5714	0	0.4286	0	
	0.8571	0	0.1429	0	0	0	0	0	
	0	0	0	0	val(:, :, 47) =	0.6154	0.0769	0.3077	0

```

0.5000    0.5000    0    0
0.5714    0    0.4286    0
0    0    0    0
val(:, :, 48) =
0.8333    0    0.1667    0
0    0    0    0
0.6667    0    0    0.3333
1.0000    0    0    0
val(:, :, 49) =
0.8333    0    0.1667    0
0    0    0    0
0.6667    0    0    0.3333
1.0000    0    0    0
val(:, :, 50) =
0.8333    0    0.1667    0
0    0    0    0
0.6667    0    0    0.3333
1.0000    0    0    0
val(:, :, 51) =
0.7059    0    0.2941    0
0    0    0    0
1.0000    0    0    0
0    0    0    0
val(:, :, 52) =
0.6667    0.0667    0.2000    0.0667
0    0    0    0
0.6667    0    0.3333    0
0    0    1.0000    0
val(:, :, 53) =
0.7143    0    0.2143    0.0714
1.0000    0    0    0
0.6667    0    0.3333    0
0    0    1.0000    0
val(:, :, 54) =
0.7333    0    0.2000    0.0667
1.0000    0    0    0
0.6000    0    0.4000    0
1.0000    0    0    0
val(:, :, 55) =
0.6875    0    0.3125    0
0    0    0    0
0.8333    0    0.1667    0
0    0    0    0
val(:, :, 56) =
0.6429    0    0.3571    0
1.0000    0    0    0
0.7143    0    0.2857    0
0    0    0    0
val(:, :, 57) =
0.6667    0    0.3333    0
0    0    0    0
0.7143    0    0.2857    0
0    0    0    0
val(:, :, 58) =
0.6429    0.0714    0.2857    0
0    0    0    1.0000
0.8333    0    0.1667    0
0    0    1.0000    0
val(:, :, 59) =
0.6667    0    0.3333    0
0    0    0    0
0.7143    0    0.2857    0
0    0    0    0
val(:, :, 60) =
0.7647    0    0.2353    0
1.0000    0    0    0
1.0000    0    0    0
0    0    0    0
val(:, :, 61) =
0.5556    0    0.4444    0
0    0.8000    0.2000    0
0.5714    0    0.2857    0.1429
1.0000    0    0    0
val(:, :, 62) =
0.5556    0    0.4444    0
0    0.8000    0.2000    0
0.5714    0    0.2857    0.1429
1.0000    0    0    0
val(:, :, 63) =
0.5556    0    0.4444    0
0    0.8000    0.2000    0
0.5714    0    0.2857    0.1429
1.0000    0    0    0
val(:, :, 64) =
0.5000    0    0.4000    0.1000
0    0.8000    0.2000    0
0.8333    0    0.1667    0
1.0000    0    0    0
val(:, :, 65) =
0.5556    0    0.3333    0.1111
0    0.8333    0.1667    0
0.6667    0    0.3333    0
1.0000    0    0    0
val(:, :, 66) =

```

0.5556	0	0.3333	0.1111	0.6000	0	0.2000	0.2000
0	0.8333	0.1667	0	1.0000	0	0	0
0.6667	0	0.3333	0				
1.0000	0	0	0				
				val(:, :, 76) =			
				0.7500	0	0.2500	0
				0	0	0	0
				0.6000	0	0.2000	0.2000
				1.0000	0	0	0
val(:, :, 67) =							
0.5000	0	0.4000	0.1000				
0.2000	0.8000	0	0				
0.6667	0	0.3333	0				
1.0000	0	0	0				
				val(:, :, 77) =			
				0.7500	0	0.2500	0
				0	0	0	0
				0.6000	0	0.2000	0.2000
				1.0000	0	0	0
val(:, :, 68) =							
0.5455	0	0.3636	0.0909				
0.2500	0.7500	0	0				
0.6667	0	0.3333	0				
1.0000	0	0	0				
				val(:, :, 78) =			
				0.7647	0	0.1765	0.0588
				0	0	0	0
				0.7500	0	0.2500	0
				1.0000	0	0	0
val(:, :, 69) =							
0.5833	0	0.3333	0.0833				
0	0.5000	0.5000	0				
0.7143	0	0.2857	0				
1.0000	0	0	0				
				val(:, :, 79) =			
				0.7500	0	0.2500	0
				0	0	0	0
				0.6000	0	0.2000	0.2000
				1.0000	0	0	0
val(:, :, 70) =							
0.5385	0	0.3846	0.0769				
1.0000	0	0	0				
0.7143	0	0.2857	0				
1.0000	0	0	0				
				val(:, :, 80) =			
				0.7647	0	0.1765	0.0588
				0	0	0	0
				0.7500	0	0.2500	0
				1.0000	0	0	0
val(:, :, 71) =							
0.6000	0	0.3000	0.1000				
0	0.5000	0.5000	0				
0.5714	0.1429	0.2857	0				
1.0000	0	0	0				
				val(:, :, 81) =			
				0.7059	0	0.2353	0.0588
				0	0	0	0
				1.0000	0	0	0
				1.0000	0	0	0
val(:, :, 72) =							
0.6000	0	0.2667	0.1333				
0	0	0	0				
0.8000	0	0.2000	0				
1.0000	0	0	0				
				val(:, :, 82) =			
				0.7647	0	0.1765	0.0588
				0	0	0	0
				0.7500	0	0.2500	0
				1.0000	0	0	0
val(:, :, 73) =							
0.6000	0	0.2667	0.1333				
0	0	0	0				
0.8000	0	0.2000	0				
1.0000	0	0	0				
				val(:, :, 83) =			
				0.7647	0	0.1765	0.0588
				0	0	0	0
				0.7500	0	0.2500	0
				1.0000	0	0	0
val(:, :, 74) =							
0.6000	0	0.2667	0.1333				
0	0	0	0				
0.8000	0	0.2000	0				
1.0000	0	0	0				
				val(:, :, 84) =			
				0.7647	0	0.1765	0.0588
				0	0	0	0
				0.7500	0	0.2500	0
				1.0000	0	0	0
val(:, :, 75) =							
0.7500	0	0.2500	0				
0	0	0	0				


```

0      0      1.0000      0
val(:, :, 104) =
0.5385      0      0.3846      0.0769
0      0      0      0
0.6250      0      0.3750      0
0      0      0      1.0000

```

```

val(:, :, 105) =
0.5000      0      0.4167      0.0833
0      0      0      0
0.5556      0      0.4444      0
0      0      1.0000      0

```

```

val(:, :, 106) =
0.5385      0      0.4615      0
0      0      0      0
0.5556      0      0.4444      0
0      0      0      0

```

T1213 - Praga 1

```

val(:, :, 1) =
0.7500      0      0.2500      0
0      0      0      0
0.8000      0      0.2000      0
0      0      0      0

```

```

val(:, :, 2) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 3) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 4) =
1      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

```

```

val(:, :, 5) =
1      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

```

```

val(:, :, 6) =
0.6875      0      0.3125      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 7) =
0.6154      0      0.3846      0
0      0      0      0
0.6250      0      0.3750      0
0      0      0      0

```

```

val(:, :, 8) =
0.5833      0.0833      0.3333      0
0      0      1.0000      0
0.6250      0      0.3750      0
0      0      0      0

```

```

val(:, :, 9) =
0.5833      0.0833      0.3333      0
0      0      0      1.0000
0.5714      0      0.4286      0
1.0000      0      0      0

```

```

val(:, :, 10) =
0.5833      0.0833      0.3333      0
0      0      1.0000      0
0.6250      0      0.3750      0
0      0      0      0

```

```

val(:, :, 11) =
0.6154      0      0.3846      0
0      0      0      0
0.6250      0      0.3750      0
0      0      0      0

```

```

val(:, :, 12) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 13) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 14) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 15) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 16) =

```

0.5455	0	0.4545	0	0	0	0	0
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 17) =							
				0.5000	0	0.5000	0
				0	0	0	0
				1.0000	0	0	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 18) =							
				0.5714	0	0.4286	0
				0	0	0	0
				0.8571	0	0.1429	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 19) =							
				0.7692	0	0.2308	0
				0	0	0	0
				0.3750	0	0.6250	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 20) =							
				0.7692	0	0.2308	0
				0	0	0	0
				0.3750	0	0.6250	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 21) =							
				0.6923	0	0.3077	0
				0	0	0	0
				0.5000	0	0.5000	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 22) =							
				0.6923	0	0.3077	0
				0	0	0	0
				0.5000	0	0.5000	0
				0	0	0	0
0.5455	0	0.4545	0				
0	0	0	0				
0.5000	0	0.5000	0				
0	0	0	0				
val(:, :, 23) =							
				0.6667	0	0.3333	0
				0	0	0	0
				0.8333	0	0.1667	0
				0	0	0	0
0.5000	0	0.5000	0				
0	0	0	0				
1.0000	0	0	0				
0	0	0	0				
val(:, :, 24) =							
				0.6667	0	0.3333	0
				0	0	0	0
				0.8333	0	0.1667	0
				0	0	0	0
0.5000	0	0.5000	0				
0	0	0	0				
1.0000	0	0	0				
0	0	0	0				
val(:, :, 25) =							
				0.6667	0	0.3333	0
				0	0	0	0
				0.8333	0	0.1667	0
				0	0	0	0
0.5000	0	0.5000	0				
0	0	0	0				
1.0000	0	0	0				
0	0	0	0				


```

val(:,:,35) =
    1.0000    0    0    0
         0    0    0    0
    0.6667    0  0.3333    0
         0    0    0    0
    0.8333    0  0.1667    0
         0    0    0    0
val(:,:,45) =
    0.7500    0  0.2500    0
         0    0    0    0
    0.8000    0  0.2000    0
         0    0    0    0
val(:,:,36) =
    0.8235    0  0.1765    0
         0    0    0    0
    0.7500    0  0.2500    0
         0    0    0    0
val(:,:,46) =
    0.7500    0  0.2500    0
         0    0    0    0
    0.8000    0  0.2000    0
         0    0    0    0
val(:,:,37) =
    0.8235    0  0.1765    0
         0    0    0    0
    0.7500    0  0.2500    0
         0    0    0    0
val(:,:,47) =
    0.7500    0  0.2500    0
         0    0    0    0
    0.8000    0  0.2000    0
         0    0    0    0
val(:,:,38) =
    0.8235    0  0.1765    0
         0    0    0    0
    0.7500    0  0.2500    0
         0    0    0    0
val(:,:,48) =
    0.7500    0  0.2500    0
         0    0    0    0
    0.8000    0  0.2000    0
         0    0    0    0
val(:,:,39) =
    0.8889    0  0.1111    0
         0    0    0    0
    0.6667    0  0.3333    0
         0    0    0    0
val(:,:,49) =
    0.5714    0  0.4286    0
         0    0    0    0
    0.8571    0  0.1429    0
         0    0    0    0
val(:,:,40) =
    0.8947    0  0.1053    0
         0    0    0    0
    1.0000    0    0    0
         0    0    0    0
val(:,:,50) =
    0.5714    0  0.4286    0
         0    0    0    0
    0.8571    0  0.1429    0
         0    0    0    0
val(:,:,41) =
    0.8333    0  0.1667    0
         0    0    0    0
    1.0000    0    0    0
         0    0    0    0
val(:,:,51) =
    0.6667    0  0.3333    0
         0    0    0    0
    0.8333    0  0.1667    0
         0    0    0    0
val(:,:,42) =
    0.8333    0  0.1667    0
         0    0    0    0
    1.0000    0    0    0
         0    0    0    0
val(:,:,52) =
    0.6667    0  0.3333    0
         0    0    0    0
    0.8333    0  0.1667    0
         0    0    0    0
val(:,:,43) =
    0.7647    0  0.2353    0
         0    0    0    0
    1.0000    0    0    0
         0    0    0    0
val(:,:,53) =
    0.6429    0  0.3571    0
    1.0000    0    0    0
    0.6667    0.1667  0.1667    0
         0    0    0    0
val(:,:,44) =
    0.8421  0.1579    0    0
         0.6154    0  0.3846    0
         1.0000    0    0    0
         0.5714  0.1429  0.2857    0
         0    0    0    0

```

```

                                0.6429      0      0.3571      0
val(:, :, 54) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0
    0.5000    0.0833    0.3333    0.0833
    1.0000      0          0          0
    0.6667    0.1667    0.1667    0
    0          0          1.0000    0

                                0.6429      0      0.3571      0
val(:, :, 64) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0

                                0.6429      0      0.3571      0
val(:, :, 55) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0
    0.7333      0      0.2667      0
    0          0          0          0
    0.6667      0      0.3333      0
    0          0          0          0

                                0.6429      0      0.3571      0
val(:, :, 56) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0
    0.6923      0      0.3077      0
    0          0          0          0
    0.5000      0      0.5000      0
    0          0          0          0

                                0.6429      0      0.3571      0
val(:, :, 57) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0
    0.6154      0      0.3846      0
    1.0000      0          0          0
    0.5714    0.1429    0.2857      0
    0          0          0          0

                                0.6667      0      0.3333      0
val(:, :, 58) =                0          0      1.0000      0
                                0.5714      0      0.2857    0.1429
                                0      1.0000      0          0
    0.6154      0      0.3846      0
    1.0000      0          0          0
    0.5714    0.1429    0.2857      0
    0          0          0          0

                                0.6429      0      0.3571      0
val(:, :, 59) =                0          0          0          0
                                0.7143      0      0.2857      0
                                0          0          0          0
    0.6154      0      0.3846      0
    1.0000      0          0          0
    0.5714    0.1429    0.2857      0
    0          0          0          0

                                0.5714      0      0.4286      0
val(:, :, 60) =                0          0          0          0
                                0.8571      0      0.1429      0
                                0          0          0          0
    0.6923      0      0.3077      0
    0          0          0          0
    0.5000      0      0.5000      0
    0          0          0          0

                                0.5000      0      0.5000      0
val(:, :, 61) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          0          0
    0.5714      0      0.4286      0
    0          0          0          0
    0.8571      0      0.1429      0
    0          0          0          0

                                0.5000      0      0.5000      0
val(:, :, 62) =                0          0          0          0
                                1.0000      0          0          0
                                0          0          0          0
    0.5714      0      0.4286      0
    0          0          0          0
    0.8571      0      0.1429      0
    0          0          0          0

                                0.3636      0      0.6364      0
val(:, :, 63) =                0          0          0          0

```

0.7000	0	0.3000	0						
0	0	0	0						
					val(:, :, 82) =				
						0.5455	0	0.4545	0
						0	0	1.0000	0
val(:, :, 73) =						0.5556	0.1111	0.3333	0
						0	0	0	0
0.3636	0	0.5455	0.0909						
0	0	0	0						
0.6667	0	0.3333	0						
1.0000	0	0	0						
					val(:, :, 83) =				
						0.6154	0	0.3846	0
						0	0	0	0
val(:, :, 74) =						0.6250	0	0.3750	0
						0	0	0	0
0.3636	0	0.5455	0.0909						
0	0	0	0						
0.6667	0	0.3333	0						
1.0000	0	0	0						
					val(:, :, 84) =				
						0.6154	0	0.3846	0
						0	0	0	0
val(:, :, 75) =						0.5714	0	0.2857	0.1429
						1.0000	0	0	0
0.6154	0	0.3846	0						
0	0	0	0						
0.6250	0	0.3750	0						
0	0	0	0						
					val(:, :, 85) =				
						0.6154	0	0.3846	0
						0	0	0	0
val(:, :, 76) =						0.5714	0	0.2857	0.1429
						1.0000	0	0	0
0.6154	0	0.3846	0						
0	0	0	0						
0.6250	0	0.3750	0						
0	0	0	0						
					val(:, :, 86) =				
						0.5714	0	0.4286	0
						0	0	0	0
val(:, :, 77) =						0.8571	0	0.1429	0
						0	0	0	0
0.6154	0	0.3846	0						
0	0	0	0						
0.5714	0	0.2857	0.1429						
1.0000	0	0	0						
					val(:, :, 87) =				
						0.6667	0	0.3333	0
						0	0	0	0
val(:, :, 78) =						0.8333	0	0.1667	0
						0	0	0	0
0.5385	0	0.4615	0						
0	0	0	0						
0.7500	0	0.2500	0						
0	0	0	0						
					val(:, :, 88) =				
						0.6429	0	0.3571	0
						0	0	0	0
val(:, :, 79) =						0.7143	0	0.2857	0
						0	0	0	0
0.4545	0	0.5455	0						
0	0	0	0						
0.6000	0	0.4000	0						
0	0	0	0						
					val(:, :, 89) =				
						0.6429	0	0.2857	0.0714
						1.0000	0	0	0
						0.8000	0	0.2000	0
						0	1.0000	0	0
0.4545	0	0.5455	0						
0	0	0	0						
0.6000	0	0.4000	0						
0	0	0	0						
					val(:, :, 90) =				
						0.6667	0	0.3333	0
						0	0	0	0
val(:, :, 81) =						0.8333	0	0.1667	0
						0	0	0	0
0.5455	0	0.4545	0						
0	0	1.0000	0						
0.5556	0.1111	0.3333	0						
0	0	0	0						
					val(:, :, 91) =				

0.6429	0	0.3571	0	0	0	0	0
0	0	0	0				
0.7143	0	0.2857	0				
0	0	0	0				
val(:, :, 92) =				val(:, :, 101) =			
				0.6364	0	0.3636	0
				0	0	0	0
0.7333	0	0.2667	0	0.4000	0	0.6000	0
0	0	0	0	0	0	0	0
0.6667	0	0.3333	0				
0	0	0	0	val(:, :, 102) =			
val(:, :, 93) =				0.6667	0	0.3333	0
				0	0	0	0
0.6667	0	0.3333	0	0.6667	0	0.3333	0
0	0	0	0	0	0	0	0
0.8333	0	0.1667	0				
0	0	0	0	val(:, :, 103) =			
val(:, :, 94) =				0.5000	0	0.5000	0
				0	0	0	0
0.7500	0	0.2500	0	0.8571	0	0.1429	0
0	0	0	0	0	0	0	0
0.8000	0	0.2000	0				
0	0	0	0	val(:, :, 104) =			
val(:, :, 95) =				0.6667	0	0.3333	0
				0	0	0	0
0.7333	0	0.2667	0	0.8333	0	0.1667	0
0	0	0	0	0	0	0	0
0.6667	0	0.3333	0				
0	0	0	0	val(:, :, 105) =			
val(:, :, 96) =				0.6667	0	0.3333	0
				0	0	0	0
0.6667	0	0.3333	0	0.8333	0	0.1667	0
0	0	0	0	0	0	0	0
0.6667	0	0.3333	0				
0	0	0	0	val(:, :, 106) =			
val(:, :, 97) =				0.7500	0	0.2500	0
				0	0	0	0
0.5000	0	0.5000	0	0.8000	0	0.2000	0
0	0	0	0	0	0	0	0
0.4545	0	0.5455	0				
0	0	0	0	<u>T1213 – Praha 2</u>			
val(:, :, 98) =				val(:, :, 1) =			
				0.7500	0	0.2500	0
				0	0	0	0
0.5000	0	0.5000	0	0.8000	0	0.2000	0
0	0	0	0	0	0	0	0
0.4545	0	0.5455	0				
0	0	0	0	val(:, :, 2) =			
val(:, :, 99) =				0.7647	0	0.2353	0
				0	0	0	0
0.5000	0	0.5000	0	1.0000	0	0	0
0	0	0	0	0	0	0	0
0.4545	0	0.5455	0				
0	0	0	0	val(:, :, 3) =			
val(:, :, 100) =				0.7647	0	0.2353	0
				0	0	0	0
0.6364	0	0.3636	0	1.0000	0	0	0
0	0	0	0	0	0	0	0
0.4000	0	0.6000	0				

```

                                0.7647      0      0.2353      0
                                0            0            0            0
val(:, :, 4) =                 1.0000      0            0            0
                                0            0            0            0
                                0            0            0            0
                                0            0            0            0

                                0.7647      0      0.2353      0
                                0            0            0            0
val(:, :, 5) =                 1.0000      0            0            0
                                0            0            0            0
                                0            0            0            0
                                0            0            0            0

                                0.7647      0      0.2353      0
                                0            0            0            0
val(:, :, 6) =                 1.0000      0            0            0
                                0            0            0            0
                                0.6875      0      0.3125      0
                                0            0            0            0
                                1.0000      0            0            0
                                0            0            0            0

                                0.4545      0.0909      0.4545      0
                                0            0            0            0
val(:, :, 7) =                 0.5000      0            0.5000      0
                                0            0            0            0
                                0.5385      0.0769      0.3846      0
                                0            0            0            0
                                0.6250      0            0.3750      0
                                0            0            0            0

                                0.4545      0.0909      0.4545      0
                                0            0            0            0
val(:, :, 8) =                 0.5000      0            0.5000      0
                                0            0            0            0
                                0.5385      0.0769      0.3846      0
                                0            0            0            0
                                0.6250      0            0.3750      0
                                0            0            0            0

                                0.4545      0.0909      0.4545      0
                                0            0            0            0
val(:, :, 9) =                 0.5000      0            0.5000      0
                                0            0            0            0
                                0.5385      0.0769      0.3077      0.0769
                                0            0            0            0
                                0.5714      0            0.4286      0
                                1.0000      0            0            0

                                0.5455      0      0.4545      0
                                0            0            0            0
val(:, :, 10) =                0.5000      0            0.5000      0
                                0            0            0            0
                                0.5385      0.0769      0.3846      0
                                0            0            0            0
                                0.6250      0            0.3750      0
                                0            0            0            0

                                0.4545      0.0909      0.4545      0
                                0            0            0            0
val(:, :, 11) =                0.5000      0            0.5000      0
                                0            0            0            0
                                0.5385      0.0769      0.3846      0
                                0            0            0            0
                                0.6250      0            0.3750      0
                                0            0            0            0

                                0.4545      0.0909      0.4545      0
                                0            0            0            0
val(:, :, 12) =                0.5000      0            0.5000      0
                                0            0            0            0
                                0.7647      0      0.2353      0
                                0            0            0            0
                                1.0000      0            0            0
                                0            0            0            0

                                0.5455      0      0.4545      0
                                0            0            0            0
val(:, :, 13) =                0.5000      0            0.5000      0

```

0	0	0	0	val(:, :, 32) =				
					0.6667	0	0.3333	0
val(:, :, 23) =					0	0	0	0
					0.8333	0	0.1667	0
0.5000	0	0.5000	0		0	0	0	0
0	0	0	0					
1.0000	0	0	0	val(:, :, 33) =				
0	0	0	0		0.6154	0	0.3846	0
					0	1.0000	0	0
val(:, :, 24) =					0.6667	0.1667	0.1667	0
					0	0	0	0
0.5000	0	0.5000	0					
0	0	0	0					
1.0000	0	0	0	val(:, :, 34) =				
0	0	0	0		0.6000	0.0667	0.3333	0
					0	0	0	0
val(:, :, 25) =					0.8333	0	0.1667	0
					0	0	0	0
0.5000	0	0.5000	0					
0	0	0	0					
1.0000	0	0	0	val(:, :, 35) =				
0	0	0	0		0.6000	0.0667	0.3333	0
					0	0	0	0
val(:, :, 26) =					0.8333	0	0.1667	0
					0	0	0	0
0.5000	0	0.5000	0					
0	0	0	0					
1.0000	0	0	0	val(:, :, 36) =				
0	0	0	0		0.8667	0	0.1333	0
					0	0.5000	0.5000	0
val(:, :, 27) =					0.7500	0	0.2500	0
					0	0	0	0
0.5714	0	0.4286	0					
0	0	0	0					
0.8571	0	0.1429	0	val(:, :, 37) =				
0	0	0	0		0.8667	0	0.1333	0
					0	0.5000	0.5000	0
val(:, :, 28) =					0.7500	0	0.2500	0
					0	0	0	0
0.7692	0	0.2308	0					
0	0	0	0					
0.3750	0	0.6250	0	val(:, :, 38) =				
0	0	0	0		0.8667	0	0.1333	0
					0	0.5000	0.5000	0
val(:, :, 29) =					0.7500	0	0.2500	0
					0	0	0	0
0.7692	0	0.2308	0					
0	0	0	0					
0.3750	0	0.6250	0	val(:, :, 39) =				
0	0	0	0		0.9375	0	0.0625	0
					0	0.5000	0.5000	0
val(:, :, 30) =					0.6667	0	0.3333	0
					0	0	0	0
0.6154	0.0769	0.3077	0					
0	0	0	0					
0.5000	0	0.5000	0	val(:, :, 40) =				
0	0	0	0		0.8824	0	0.1176	0
					0.5000	0.5000	0	0
val(:, :, 31) =					1.0000	0	0	0
					0	0	0	0
0.6154	0.0769	0.3077	0					
0	0	0	0					
0.5000	0	0.5000	0	val(:, :, 41) =				
0	0	0	0		0.8333	0	0.1667	0

```

0      0      0      0
1.0000 0      0      0
0      0      0      0
val(:, :, 42) =
0.8333 0      0.1667 0
0      0      0      0
1.0000 0      0      0
0      0      0      0
val(:, :, 43) =
0.7647 0      0.2353 0
0      0      0      0
1.0000 0      0      0
0      0      0      0
val(:, :, 44) =
1 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
val(:, :, 45) =
0.6250 0.1250 0.2500 0
0.1250 0.6250 0.2500 0
0.4000 0.4000 0.2000 0
0      0      0      0
val(:, :, 46) =
0.3333 0.3333 0.3333 0
0.1000 0.7000 0.2000 0
0.6000 0.2000 0.2000 0
0      0      0      0
val(:, :, 47) =
0.4286 0.1429 0.4286 0
0.1111 0.7778 0.1111 0
0.8000 0      0.2000 0
0      0      0      0
val(:, :, 48) =
0.4545 0      0.5455 0
0.3333 0.6667 0      0
0.7143 0.1429 0.1429 0
0      0      0      0
val(:, :, 49) =
0.4545 0      0.5455 0
0.3333 0.6667 0      0
0.7143 0.1429 0.1429 0
0      0      0      0
val(:, :, 50) =
0.5455 0      0.4545 0
0.3333 0.6667 0      0
0.5714 0.2857 0.1429 0
0      0      0      0
val(:, :, 51) =
0.5833 0      0.4167 0
0.3333 0.6667 0      0
0.8333 0      0.1667 0
0      0      0      0
val(:, :, 52) =
0.4444 0.1111 0.4444 0
0.3333 0.5000 0.1667 0
0.6667 0.1667 0.1667 0
0      0      0      0
val(:, :, 53) =
0.4444 0      0.5556 0
0.4000 0.6000 0      0
0.5714 0.1429 0.2857 0
0      0      0      0
val(:, :, 54) =
0.5000 0      0.4000 0.1000
0.2500 0.7500 0      0
0.6667 0.1667 0.1667 0
0      0      1.0000 0
val(:, :, 55) =
0.7273 0      0.2727 0
0.2500 0.5000 0.2500 0
0.3333 0.3333 0.3333 0
0      0      0      0
val(:, :, 56) =
0.6667 0      0.3333 0
0.2500 0.5000 0.2500 0
0.2500 0.2500 0.5000 0
0      0      0      0
val(:, :, 57) =
0.5455 0      0.4545 0
0.3333 0.6667 0      0
0.7143 0      0.2857 0
0      0      0      0
val(:, :, 58) =
0.5000 0      0.5000 0
0.5000 0.5000 0      0
0.5714 0.1429 0.2857 0
0      0      0      0
val(:, :, 59) =
0.5455 0      0.4545 0
0.3333 0.6667 0      0
0.7143 0      0.2857 0
0      0      0      0
val(:, :, 60) =
0.5455 0      0.4545 0
0.3333 0.6667 0      0
0.5714 0.2857 0.1429 0
0      0      0      0

```

				0.8571	0	0.1429	0
	0.6923	0	0.3077	0	0	0	0
	0	0	0				
	0.5000	0	0.5000				
	0	0	0				
val(:, :, 61) =				0.4286	0.0714	0.5000	0
				0	0	0	0
				1.0000	0	0	0
				0	0	0	0
	0.5000	0.0714	0.4286				
	0	0	0				
	0.8571	0	0.1429				
	0	0	0				
val(:, :, 62) =				0.4286	0.0714	0.5000	0
				0	0	0	0
				1.0000	0	0	0
				0	0	0	0
	0.3636	0.0909	0.5455				
	0.3333	0.6667	0				
	0.8571	0	0.1429				
	0	0	0				
val(:, :, 63) =				0.3636	0	0.6364	0
				0	0	0	0
				0.7000	0	0.3000	0
				0	0	0	0
	0.5000	0.0833	0.4167				
	0.5000	0.5000	0				
	0.7143	0	0.2857				
	0	0	0				
val(:, :, 64) =				0.3636	0	0.5455	0.0909
				0	0	0	0
				0.6667	0	0.3333	0
				1.0000	0	0	0
	0.5000	0.0833	0.4167				
	0.5000	0.5000	0				
	0.7143	0	0.2857				
	0	0	0				
val(:, :, 65) =				0.3636	0	0.5455	0.0909
				0	0	0	0
				0.6667	0	0.3333	0
				1.0000	0	0	0
	0.5455	0	0.4545				
	0.3333	0.6667	0				
	0.7143	0	0.2857				
	0	0	0				
val(:, :, 66) =				0.4545	0.0909	0.4545	0
				0.5000	0.5000	0	0
				0.6250	0	0.3750	0
				0	0	0	0
	0.4000	0.1000	0.5000				
	0.5000	0.5000	0				
	0.7143	0	0.2857				
	0	0	0				
val(:, :, 67) =				0.5833	0.0833	0.3333	0
				0	0	1.0000	0
				0.5000	0.1250	0.3750	0
				0	0	0	0
	0.2500	0.1250	0.6250				
	0.2000	0.8000	0				
	0.5714	0	0.2857	0.1429			
	1.0000	0	0	0			
val(:, :, 68) =				0.5385	0.0769	0.3846	0
				0	0	0	0
				0.5714	0	0.2857	0.1429
				1.0000	0	0	0
	0.4000	0.1000	0.5000				
	0.5000	0.5000	0				
	0.7143	0	0.2857				
	0	0	0				
val(:, :, 69) =				0.4615	0.0769	0.4615	0
				0	0	0	0
				0.7500	0	0.2500	0
				0	0	0	0
	0.5000	0.0714	0.4286				
	0	0	0				


```

                                0.5714    0.0714    0.3571    0
                                0          0          0          0
val(:, :, 79) =                0.7143    0          0.2857    0
                                0          0          0          0
    0.4000    0.1000    0.5000    0
    0          0          1.0000    0
    0.5000    0.1000    0.4000    0
    0          0          0          0

                                0.6000    0.0667    0.2667    0.0667
                                0          0          0          0
val(:, :, 80) =                0.8000    0          0.2000    0
                                1.0000    0          0          0
    0.3636    0.0909    0.5455    0
    0          0          0          0
    0.6000    0          0.4000    0
    0          0          0          0

                                0.6000    0.0667    0.3333    0
                                0          0          0          0
val(:, :, 81) =                0.8333    0          0.1667    0
                                0          0          0          0
    0.4167    0.0833    0.5000    0
    0          0          0          0
    0.6667    0          0.3333    0
    0          0          0          0

                                0.5714    0.0714    0.3571    0
                                0          0          0          0
val(:, :, 82) =                0.7143    0          0.2857    0
                                0          0          0          0
    0.4545    0.0909    0.4545    0
    0          0          1.0000    0
    0.5556    0.1111    0.3333    0
    0          0          0          0

                                0.7333    0          0.2667    0
                                0          0          0          0
val(:, :, 83) =                0.6667    0          0.3333    0
                                0          0          0          0
    0.5000    0.1667    0.3333    0
    0          0          1.0000    0
    0.6250    0          0.3750    0
    0          0          0          0

                                0.6000    0.0667    0.3333    0
                                0          0          0          0
val(:, :, 84) =                0.8333    0          0.1667    0
                                0          0          0          0
    0.5455    0.1818    0.2727    0
    0          0          1.0000    0
    0.4286    0.1429    0.2857    0.1429
    1.0000    0          0          0

                                0.7500    0          0.2500    0
                                0          0          0          0
val(:, :, 85) =                0.8000    0          0.2000    0
                                0          0          0          0
    0.5385    0.0769    0.3846    0
    0          0          0          0
    0.5714    0          0.2857    0.1429
    1.0000    0          0          0

                                0.7333    0          0.2667    0
                                0          0          0          0
val(:, :, 86) =                0.6667    0          0.3333    0
                                0          0          0          0
    0.4167    0.0833    0.5000    0
    0.5000    0.5000    0          0
    0.8571    0          0.1429    0
    0          0          0          0

                                0.6667    0          0.3333    0
                                0          0          0          0
val(:, :, 87) =                0.6667    0          0.3333    0
                                0          0          0          0
    0.6000    0.0667    0.3333    0
    0          0          0          0
    0.8333    0          0.1667    0
    0          0          0          0

                                0.3750    0          0.6250    0
                                0          1.0000    0          0
val(:, :, 88) =                0.3636    0.0909    0.5455    0

```

```

0      0      0      0
val(:, :, 98) =
0.3750      0      0.6250      0
0      1.0000      0      0
0.3636      0.0909      0.5455      0
0      0      0      0

```

```

val(:, :, 99) =
0.3750      0      0.6250      0
0      1.0000      0      0
0.3636      0.0909      0.5455      0
0      0      0      0

```

```

val(:, :, 100) =
0.5455      0.0909      0.3636      0
0      0      0      0
0.4000      0      0.6000      0
0      0      0      0

```

```

val(:, :, 101) =
0.5455      0.0909      0.3636      0
0      0      0      0
0.4000      0      0.6000      0
0      0      0      0

```

```

val(:, :, 102) =
0.6667      0      0.3333      0
0      0      0      0
0.6667      0      0.3333      0
0      0      0      0

```

```

val(:, :, 103) =
0.5000      0      0.5000      0
0      0      0      0
0.8571      0      0.1429      0
0      0      0      0

```

```

val(:, :, 104) =
0.6667      0      0.3333      0
0      0      0      0
0.8333      0      0.1667      0
0      0      0      0

```

```

val(:, :, 105) =
0.6667      0      0.3333      0
0      0      0      0
0.8333      0      0.1667      0
0      0      0      0

```

```

val(:, :, 106) =
0.7500      0      0.2500      0
0      0      0      0
0.8000      0      0.2000      0
0      0      0      0

```

T1213 – Praga 3

```

val(:, :, 1) =
0.7333      0.0667      0.2000      0
0      0      1.0000      0
0.8000      0      0.2000      0
0      0      0      0

```

```

val(:, :, 2) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 3) =
0.7647      0      0.2353      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 4) =
1      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

```

```

val(:, :, 5) =
1      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

```

```

val(:, :, 6) =
0.6875      0      0.3125      0
0      0      0      0
1.0000      0      0      0
0      0      0      0

```

```

val(:, :, 7) =
0.5385      0.0769      0.3846      0
0      0      0      0
0.6250      0      0.3750      0
0      0      0      0

```

```

val(:, :, 8) =
0.5833      0      0.4167      0
0      1.0000      0      0
0.5000      0.1250      0.3750      0
0      0      0      0

```

```

val(:, :, 9) =
0.5833      0      0.3333      0.0833
0      1.0000      0      0
0.4286      0.1429      0.4286      0
1.0000      0      0      0

```

```

val(:, :, 10) =

```

	0.5833	0	0.4167	0	0.4000	0.1000	0.5000	0
	0	1.0000	0	0	0	0	0	0
	0.5000	0.1250	0.3750	0				
	0	0	0	0				
val(:, :, 11) =					0.5000	0	0.5000	0
	0.5455	0.0909	0.3636	0	0	1.0000	0	0
	0	0.5000	0.5000	0	0.4000	0.1000	0.5000	0
	0.5000	0.1250	0.3750	0	0	0	0	0
	0	0	0	0				
val(:, :, 12) =					0.4444	0.1111	0.4444	0
	0.7500	0	0.2500	0	0	0.5000	0.5000	0
	1.0000	0	0	0	0.4000	0.1000	0.5000	0
	0.7500	0.2500	0	0	0	0	0	0
	0	0	0	0				
val(:, :, 13) =					0.4286	0	0.5714	0
	0.7647	0	0.2353	0	0	0.7500	0.2500	0
	0	0	0	0	0.4000	0.1000	0.5000	0
	1.0000	0	0	0	0	0	0	0
	0	0	0	0				
val(:, :, 14) =					0.4615	0	0.5385	0
	0.7500	0	0.2500	0	1.0000	0	0	0
	1.0000	0	0	0	0.8571	0.1429	0	0
	0.7500	0.2500	0	0	0	0	0	0
	0	0	0	0				
val(:, :, 15) =					0.4615	0	0.5385	0
	0.7059	0.0588	0.2353	0	0	1.0000	0	0
	0	0	0	0	0.8571	0.1429	0	0
	1.0000	0	0	0	0	0	0	0
	0	0	0	0				
val(:, :, 16) =					0.3333	0.0833	0.5833	0
	0.4286	0	0.5714	0	0.5000	0.5000	0	0
	0	0.7500	0.2500	0	0.8571	0.1429	0	0
	0.4000	0.1000	0.5000	0	0	0	0	0
	0	0	0	0				
val(:, :, 17) =					0.4615	0	0.5385	0
	0.4444	0.1111	0.4444	0	0	1.0000	0	0
	0	0.5000	0.5000	0	0.8571	0.1429	0	0
	0.4000	0.1000	0.5000	0	0	0	0	0
	0	0	0	0				
val(:, :, 18) =					0.4167	0.0833	0.5000	0
	0.5000	0	0.5000	0	0.5000	0.5000	0	0
	0	1.0000	0	0	0.7143	0.1429	0.1429	0
	0.4000	0.1000	0.5000	0	0	0	0	0
	0	0	0	0				
val(:, :, 19) =					0.7500	0	0.2500	0
	0.5000	0	0.5000	0	0	1.0000	0	0
	0	1.0000	0	0	0.2500	0.1250	0.6250	0
	0.5000	0	0.5000	0	0	0	0	0
	0	1.0000	0	0				

```

                                0.8235      0      0.1765      0
                                0            0            0            0
val(:, :, 29) =                0.7500      0      0.2500      0
                                0            0            0            0
                                0.2500      0.1250      0.6250      0
                                0            0            0            0

                                0.8889      0      0.1111      0
                                0            0            0            0
val(:, :, 30) =                0.6667      0      0.3333      0
                                0            0            0            0

                                0.8947      0      0.1053      0
                                0            0            0            0
val(:, :, 31) =                1.0000      0            0            0
                                0            0            0            0

                                0.8333      0      0.1667      0
                                0            0            0            0
val(:, :, 32) =                1.0000      0            0            0
                                0            0            0            0
                                0.3333      0      0.6667      0
                                0.1111      0.7778      0.1111      0
                                0.5000      0.3333      0.1667      0
                                0            0            0            0

                                0.8333      0      0.1667      0
                                0            0            0            0
val(:, :, 33) =                1.0000      0            0            0
                                0            0            0            0
                                0.2857      0      0.7143      0
                                0.2500      0.7500      0            0
                                0.5000      0.3333      0.1667      0
                                0            0            0            0

                                0.7647      0      0.2353      0
                                0            0            0            0
val(:, :, 34) =                1.0000      0            0            0
                                0            0            0            0
                                0.3333      0.2222      0.4444      0
                                0.1667      0.6667      0.1667      0
                                0.6667      0.1667      0.1667      0
                                0            0            0            0

                                0.6154      0.3846      0            0
                                0.6250      0.3750      0            0
val(:, :, 35) =                0            0            0            0
                                0            0            0            0
                                0.4545      0.1818      0.3636      0
                                0.2500      0.5000      0.2500      0
                                0.6667      0.1667      0.1667      0
                                0            0            0            0

                                0.6154      0.0769      0.3077      0
                                0.3333      0.6667      0            0
val(:, :, 36) =                0.8000      0            0.2000      0
                                0            0            0            0
                                0.8235      0      0.1765      0
                                0            0            0            0
                                0.7500      0      0.2500      0
                                0            0            0            0

                                0.6667      0.0667      0.2667      0
                                1.0000      0            0            0
val(:, :, 37) =                0.8000      0            0.2000      0
                                0            0            0            0
                                0.8235      0      0.1765      0
                                0            0            0            0
                                0.7500      0      0.2500      0
                                0            0            0            0

                                0.6667      0.0667      0.2667      0
                                1.0000      0            0            0
val(:, :, 38) =                0.8000      0            0.2000      0

```



```

      0      0.8750      0.1250      0
      0.5714      0.1429      0.2857      0
      0      0      0      0
val(:, :, 67) =
      0.3000      0.3000      0.4000      0
      0.3333      0.3333      0.3333      0
      0.5714      0      0.2857      0.1429
      1.0000      0      0      0
val(:, :, 68) =
      0.4000      0.2000      0.4000      0
      0.2500      0.5000      0.2500      0
      0.5714      0.1429      0.2857      0
      0      0      0      0
val(:, :, 69) =
      0.3636      0.1818      0.4545      0
      0.3333      0.3333      0.3333      0
      0.7143      0.1429      0.1429      0
      0      0      0      0
val(:, :, 70) =
      0.2500      0.2500      0.5000      0
      0.5000      0      0.5000      0
      1.0000      0      0      0
      0      0      0      0
val(:, :, 71) =
      0.2727      0.0909      0.6364      0
      0.3333      0.6667      0      0
      0.8571      0.1429      0      0
      0      0      0      0
val(:, :, 72) =
      0.3636      0      0.6364      0
      0      0      0      0
      0.7000      0      0.3000      0
      0      0      0      0
val(:, :, 73) =
      0.3636      0      0.5455      0.0909
      0      0      0      0
      0.6667      0      0.3333      0
      1.0000      0      0      0
val(:, :, 74) =
      0.3636      0      0.5455      0.0909
      0      0      0      0
      0.6667      0      0.3333      0
      1.0000      0      0      0
val(:, :, 75) =
      0.6154      0      0.3846      0
      0      0      0      0
      0.6250      0      0.3750      0
      0      0      0      0
val(:, :, 76) =
      0.5385      0.0769      0.3846      0
      0      0      0      0
      0.6250      0      0.3750      0
      0      0      0      0
val(:, :, 77) =
      0.6154      0      0.3846      0
      0      0      0      0
      0.5714      0      0.2857      0.1429
      1.0000      0      0      0
val(:, :, 78) =
      0.5385      0      0.4615      0
      0      0      0      0
      0.7500      0      0.2500      0
      0      0      0      0
val(:, :, 79) =
      0.4000      0.1000      0.5000      0
      0      0      1.0000      0
      0.6000      0      0.4000      0
      0      0      0      0
val(:, :, 80) =
      0.4545      0      0.5455      0
      0      0      0      0
      0.6000      0      0.4000      0
      0      0      0      0
val(:, :, 81) =
      0.5000      0      0.5000      0
      0      0      0      0
      0.6667      0      0.3333      0
      0      0      0      0
val(:, :, 82) =
      0.5000      0      0.5000      0
      0      0      0      0
      0.6667      0      0.3333      0
      0      0      0      0
val(:, :, 83) =
      0.6154      0      0.3846      0
      0      0      0      0
      0.6250      0      0.3750      0
      0      0      0      0
val(:, :, 84) =
      0.6154      0      0.3846      0
      0      0      0      0
      0.5714      0      0.2857      0.1429
      1.0000      0      0      0
val(:, :, 85) =

```

				0.8000	0	0.2000	0
	0.6154	0	0.3846	0	0	0	0
	0	0	0				
	0.5714	0	0.2857	0.1429			
	1.0000	0	0	0			
				val(:, :, 95) =			
				0.7333	0	0.2667	0
				0	0	0	0
val(:, :, 86) =				0.6667	0	0.3333	0
				0	0	0	0
	0.5714	0	0.4286	0	0	0	0
	0	0	0				
	0.8571	0	0.1429				
	0	0	0				
				val(:, :, 96) =			
				0.6667	0	0.3333	0
				0	0	0	0
val(:, :, 87) =				0.6667	0	0.3333	0
				0	0	0	0
	0.6667	0	0.3333	0	0	0	0
	0	0	0				
	0.8333	0	0.1667				
	0	0	0				
				val(:, :, 97) =			
				0.5000	0	0.5000	0
				0	0	0	0
val(:, :, 88) =				0.4545	0	0.5455	0
				0	0	0	0
	0.6429	0	0.3571				
	0	0	0				
	0.7143	0	0.2857				
	0	0	0				
				val(:, :, 98) =			
				0.4000	0.1000	0.5000	0
				0	0	0	0
val(:, :, 89) =				0.4545	0	0.5455	0
				0	0	0	0
	0.6667	0	0.2667	0.0667			
	0	0	0				
	0.8000	0	0.2000				
	1.0000	0	0				
				val(:, :, 99) =			
				0.4000	0.1000	0.5000	0
				0	0	0	0
val(:, :, 90) =				0.4545	0	0.5455	0
				0	0	0	0
	0.6667	0	0.3333				
	0	0	0				
	0.8333	0	0.1667				
	0	0	0				
				val(:, :, 100) =			
				0.6364	0	0.3636	0
				0	0	0	0
val(:, :, 91) =				0.4000	0	0.6000	0
				0	0	0	0
	0.6429	0	0.3571				
	0	0	0				
	0.7143	0	0.2857				
	0	0	0				
				val(:, :, 101) =			
				0.6364	0	0.3636	0
				0	0	0	0
val(:, :, 92) =				0.4000	0	0.6000	0
				0	0	0	0
	0.7333	0	0.2667				
	0	0	0				
	0.6667	0	0.3333				
	0	0	0				
				val(:, :, 102) =			
				0.6667	0	0.3333	0
				0	0	0	0
val(:, :, 93) =				0.6667	0	0.3333	0
				0	0	0	0
	0.6667	0	0.3333				
	0	0	0				
	0.8333	0	0.1667				
	0	0	0				
				val(:, :, 103) =			
				0.5000	0	0.5000	0
				0	0	0	0
val(:, :, 94) =				0.8571	0	0.1429	0
				0	0	0	0
	0.7500	0	0.2500				
	0	0	0				

```
                                0      0      0      0
                                0.8333  0      0.1667  0
                                0      0      0      0
val(:, :, 104) =
    0.6667    0    0.3333    0
         0      0      0      0
    0.8333    0    0.1667    0
         0      0      0      0

val(:, :, 105) =
    0.6667    0    0.3333    0

                                0      0      0      0
                                0.7500  0      0.2500  0
                                0      0      0      0
val(:, :, 106) =
    0.8000    0    0.2000    0
         0      0      0      0
```


APÊNDICE D

MODELO GERAL E MÉDIA DAS MATRIZES ENCONTRADAS

Modelo geral

M_T1011_BMinP =

0.6869	0.0040	0.2623	0.0467
0.4091	0.4394	0.1364	0.0152
0.7058	0.0140	0.2715	0.0088
0.5714	0.1209	0.2418	0.0659

M_T1213_AcVermP =

0.6110	0.0369	0.3485	0.0036
0.2439	0.6016	0.1545	0
0.6639	0.0323	0.2982	0.0056
0.8889	0	0.1111	0

M_T1112_BMinP =

0.6682	0.0151	0.2814	0.0353
0.4320	0.2800	0.0480	0.2400
0.7016	0.0332	0.2321	0.0332
0.1792	0.6415	0.0755	0.1038

M_T1011_FerrP =

0.6113	0.0811	0.2607	0.0469
0.1148	0.5926	0.2500	0.0426
0.4606	0.2592	0.2715	0.0088
0.3956	0.2967	0.2418	0.0659

M_T1213_BMinP =

0.6597	0.0047	0.3329	0.0027
0.6250	0	0.3125	0.0625
0.6850	0.0113	0.2982	0.0056
0.6667	0.2222	0.1111	0

M_T1112_FerrP =

0.6420	0.0180	0.2827	0.0574
0.2427	0.6214	0.1311	0.0049
0.7155	0.0192	0.2321	0.0332
0.8208	0	0.0755	0.1038

M_T1011_AcVermP =

0.5825	0.0223	0.3301	0.0650
0.1161	0.7603	0.1161	0.0075
0.5832	0.1366	0.2715	0.0088
0.5934	0.0989	0.2418	0.0659

M_T1213_FerrP =

0.6127	0.0331	0.3505	0.0037
0.1892	0.6419	0.1689	0
0.6442	0.0520	0.2982	0.0056
0.8889	0	0.1111	0

M_T1112_AcVermP =

0.5915	0.0249	0.3314	0.0522
0.1726	0.6996	0.0807	0.0471
0.6440	0.0908	0.2321	0.0332
0.1887	0.6321	0.0755	0.1038

Média dos modelos individuais

Media_T1011_BMin =

0.6796	0.0041	0.2690	0.0473
0.1871	0.0698	0.0544	0.0094
0.7258	0.0133	0.2345	0.0076
0.4355	0.0849	0.1714	0.0252

Media_T1112_BMin =

0.6637	0.0145	0.2856	0.0362
0.3546	0.1344	0.0267	0.1635
0.7204	0.0338	0.2088	0.0370
0.1245	0.5255	0.0755	0.0481

Media_T1213_BMin =

0.6433	0.0046	0.3489	0.0032
0.0755	0	0.0472	0.0094
0.7005	0.0106	0.2551	0.0054
0.0566	0.0189	0.0094	0

Media_T1011_AcVerm =

0.5507	0.0250	0.3513	0.0730
0.1731	0.6094	0.0987	0.0057
0.5993	0.1398	0.2345	0.0076
0.4497	0.0708	0.1714	0.0252

Media_T1112_AcVerm =

0.5727	0.0246	0.3433	0.0594
0.1961	0.6018	0.0870	0.0774
0.6717	0.0825	0.2088	0.0370
0.0830	0.5670	0.0755	0.0481

Media_T1213_AcVerm =

0.5823	0.0425	0.3713	0.0040
0.0819	0.1958	0.0808	0
0.6801	0.0311	0.2551	0.0054
0.0755	0	0.0094	0

Media_T1011_Ferr =

0.5409	0.1001	0.3150	0.0440
0.1185	0.4432	0.2131	0.0270
0.4906	0.2485	0.2345	0.0076
0.3066	0.2138	0.1714	0.0252

Media_T1112_Ferr =

0.6269	0.0199	0.2928	0.0604
0.2273	0.2940	0.0541	0.0094
0.7290	0.0252	0.2088	0.0370
0.6500	0	0.0755	0.0481

Media_T1213_Ferr =

0.5738	0.0384	0.3838	0.0039
0.1191	0.2395	0.0942	0
0.6617	0.0494	0.2551	0.0054
0.0755	0	0.0094	0

Desvio_T1011_BMin =

0.0841	0.0173	0.0909	0.0373
0.3676	0.2045	0.2146	0.0971
0.2257	0.0485	0.2124	0.0346
0.4767	0.2625	0.3567	0.1049

Desvio_T1112_BMin =

0.0812	0.0327	0.0853	0.0509
0.4065	0.2441	0.1304	0.2918
0.1681	0.0734	0.1571	0.0837
0.3138	0.4727	0.2654	0.1581

Desvio_T1213_BMin =

0.1205	0.0219	0.1194	0.0162
0.2654	0	0.2130	0.0971
0.2128	0.0378	0.1747	0.0274
0.2322	0.1367	0.0971	0

Desvio_T1011_AcVerm =

0.1262	0.0524	0.1288	0.0644
0.2400	0.3015	0.1375	0.0290
0.2196	0.1242	0.2124	0.0346
0.4760	0.2434	0.3567	0.1049

Desvio_T1112_AcVerm =

0.1202	0.0424	0.1084	0.0963
0.2259	0.3054	0.1922	0.1961
0.1947	0.0944	0.1571	0.0837
0.2459	0.4710	0.2654	0.1581

Desvio_T1213_AcVerm =

0.1606	0.0549	0.1391	0.0183
0.1608	0.3079	0.2327	0
0.2234	0.0757	0.1747	0.0274
0.2654	0	0.0971	0

Desvio_T1011_Ferr =

0.1964	0.1148	0.1997	0.0531
0.2175	0.3305	0.2398	0.0597
0.3035	0.2623	0.2124	0.0346
0.4607	0.3888	0.3567	0.1049

Desvio_T1112_Ferr =

0.1080	0.0373	0.0903	0.0591
0.3395	0.3359	0.1287	0.0971
0.1790	0.1132	0.1571	0.0837
0.4565	0	0.2654	0.1581

Desvio_T1213_Ferr =

0.1738	0.0717	0.1569	0.0178
0.2816	0.3718	0.2337	0
0.2143	0.0803	0.1747	0.0274
0.2654	0	0.0971	0

ANEXO A

EXEMPLO DE PLANILHA DE MANEJO ECOLÓGICO

Planilha com coleta de dados referente ao TB01 – setembro01 – 2012.

MANEJO ECOLÓGICO DE PRAGAS - TB01										
SETOR	A	B	C	D	E	F				
ÁREA	8,53	17,96	14,61	17,52	18,63	3,37				
Controle de Pragas										
DATA	01/set	01/set	01/set	01/set	01/set	01/set				
FITOSANITARISTA	I									
Diversos										
DATA										
FITOSANITARISTA										
Setor	Qds	Área	PRAGA 1	PRAGA 2	PRAGA 3	PRAGA 4	PRAGA 5	PRAGA 6	PRAGA 7	PRAGA 8
A	1	1,31	2		9	4	5		1	
	5	1,81	1		5	5	2		0	
	6	1,91	0		6	2	3		0	
	7	2,00	0		4	7	2		2	
	8	1,50	3		2	9	6		0	
B	9	1,73	2		5	6	3		3	
	10	2,50	0		5	4	2		0	
	11	2,30	0		4	5	3		0	
	12	2,10	0		6	4	4		0	
	13	1,68	3		4	2	5		0	
	14	2,48	2		4	1	2		0	
	15	2,85	1		2	3	3		1	
	16	2,32	0		3	3	4		0	
C	17	2,66	1		5	4	6		0	
	18	2,77	3		5	3	5		0	
	19	1,78	1		6	1	4		0	
	20	0,04	3		4	1	4		0	
	21	0,75	0		7	1	3		0	
	22	2,75	2		1	0	4		0	
	27	3,06	0		5	6	3		2	
	28	0,80	0		3	6	4		1	
D	23	6,53	0		1	7	5		0	
	24	2,88	2		4	5	6		0	
	25	1,86	2		2	6	4		0	
	26	3,18	2		5	8	3		1	
	29	3,09	0		7	4	4		0	
	30	1,39	0		4	3	2		0	
	31	1,95	3		6	4	5		0	
E	32	2,83	1		1	1	5		1	
	33	2,65	0		3	5	7		0	
	34	3,28	1		5	4	4		0	
	35	2,50	2		6	3	6		0	
	38	2,65	1		5	5	5		4	
	37	1,38	1		8	6	7		0	
F	2	1,41	2		8	0	1		0	
	3	1,10	4		9	0	3		0	
	4	0,86	2		23	0	2		0	
Nome	Data	SETOR	PRAGA 1	PRAGA 2	PRAGA 3	PRAGA 4	PRAGA 5	PRAGA 6	PRAGA 7	PRAGA 8
	1-set	A	2,0	#DIV/0!	8,7	6,8	3,6	#DIV/0!	0,6	#DIV/0!
	1-set	B	1,7	#DIV/0!	6,9	4,4	3,3	#DIV/0!	0,5	#DIV/0!
	1-set	C	2,1	#DIV/0!	7,5	3,4	4,1	#DIV/0!	0,4	#DIV/0!
	1-set	D	2,0	#DIV/0!	6,3	7,5	4,4	#DIV/0!	0,2	#DIV/0!
	1-set	E	1,9	#DIV/0!	7,9	4,8	5,1	#DIV/0!	0,6	#DIV/0!
	1-set	F	4,4	#DIV/0!	22,2	0,0	2,0	#DIV/0!	0,0	#DIV/0!