Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

Análise de Padrões de Mobilidade Utilizando Redes de Autômatos Estocásticos

Fábio Longaray Delamare

Dissertação apresentada como requisito parcial à obtenção do grau de mestre em Ciência da Computação

Orientador: Prof. Dr. Fernando Luís Dotti

Porto Alegre
2008

PUCRS
Campus Central
Av. Ipiranga, 6681 – prédio 16 – CEP 90619-900
Porto Alegre – RS – Brasil
Fone: +55 (51) 3320-3544 – Fax: +55 (51) 3320-3548
Email: bceadm@pucrs.br
www.pucrs.br/biblioteca

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada ***"Análise de Padrões de Mobilidade Utilizando Redes de Autômatos Estocásticos"***, apresentada por Fábio Longaray Delamare, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Processamento Paralelo e Distribuído, aprovada em 02/03/2007 pela Comissão Examinadora:

Prof. Dr. Fernando Luís Dotti – Orientador — PPGCC/PUCRS

Prof. Dr. Paulo Henrique Lemelle Fernandes – PPGCC/PUCRS

Prof. Dr. Lisandro Zambenedetti Granville– UFRGS

Homologada em 20/05/08, conforme Ata No. 010/08 pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P. 16 – sala 106 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@inf.pucrs.br
www.pucrs.br/facin/pos

Resumo

Muitos trabalhos sobre a análise de desempenho de Redes Ad Hoc podem ser encontrados na literatura. Estas análises podem ser através de ferramentas de simulação ou por métodos analíticos, como formalismos markovianos ou sistemas de equações. A movimentação é um fator determinante no desempenho de Redes Ad Hoc e, para possibilitar esta análise, foram criados padrões de movimentação para modelar o comportamento dos dispositivos de rede (os nodos). Neste trabalho são descritos alguns padrões de movimentação existentes na literatura, sendo os padrões Random Waypoint e Random Direction estudados em mais detalhes. São propostos modelos para representar e analisar estes dois padrões, utilizando o formalismo markoviano de Redes de Autômatos Estocásticos, possibilitando uma análise detalhada da Distribuição Espacial de Nodos de acordo com as possíveis variações destes padrões. Algumas das variações considerados nesta modelagem: velocidade, tempo de pausa, tamanho dos movimentos, escolha de caminho. A consistência destes modelos é avaliada comparando os resultados obtidos com outros resultados da literatura, demonstrando um satisfatório grau de proximidade entre os resultados.

Abstract

Several studies about the performance analysis of Ad Hoc Networks can be found in the literature. These analyses use simulation tools or analitycal methods, such Markovian models or equation systems. The mobility is a deterministic factor in the performance of Ad Hoc Networks and, to make possible these analyses, several mobility models are proposed in the literature. In this work some existing mobility models found in the literature are described, and the Random Waypoint and Random Direction models are studied in more detail. Models are proposed to represent and analyze these two mobility models, using the Markovian formalism of Stochastic Automata Networks, making possible a detailed analysis of the Spacial Node Distribution in accordance to the possible variations of these mobility models. Some of the variations considered in this modeling are: speed, pause time, size of moves, routing strategies. The consistency of the results achieved is evaluated comparing with other results from the literature, demonstrating a satisfactory degree of precision.

Lista de Figuras

Figura 1 Atividades desenvolvidas na condução da pesquisa 21
Figura 2 Exemplo de um modelo SAN. 23
Figura 3 Cadeia de Markov Equivalente. 25
Figura 4 Exemplo de movimentação gerada no padrão Random Walk. 28
Figura 5 Movimentação no Random Walk. 29
Figura 6 Exemplo de movimentação gerada pelo padrão Random Walk, usando baixos valores para distância ou tempo de movimentação. 29
Figura 7 Exemplo de movimentação gerada pela versão probabilística do Random Walk. 30
Figura 8 Movimentação no Random Waypoint. 31
Figura 9 Exemplo de movimentação gerada pelo padrão Random Waypoint. . . . 31
Figura 10 Variação do número de vizinhos no Random Waypoint. 32
Figura 11 Exemplo de movimentação gerada pelo padrão Random Direction. . . . 32
Figura 12 Movimentação no Random Direction. 33
Figura 13 Exemplo de movimentação gerada pelo Padrão de Mobilidade com Área de Movimentação sem Fronteiras. 34
Figura 14 Área de movimentação redesenhada para uma área sem fronteiras. . . . 34
Figura 15 Movimentação no Gauss-Markov. 35
Figura 16 Exemplo de movimentação gerada no Gauss-Markov. 35
Figura 17 Exemplo de movimentação gerada no City Section. 36
Figura 18 Exemplo de mapa de movimentação para o padrão de Movimentação Freeway. 37
Figura 19 Coluna de pontos de referência do padrão de movimentação em Colunas. 38
Figura 20 Movimentação dos nodos no padrão de movimentação em colunas. . . . 38
Figura 21 Exemplo de movimentação gerado no padrão de movimentação em colunas. 39
Figura 22 Movimentação de Comunidades Nômades. 40
Figura 23 Padrão de Movimentação em Perseguição. 40
Figura 24 Exemplo de comportamento da movimentação dos nodos de dois grupos diferentes no RPGM. 41
Figura 25 Modelo SAN para Random Waypoint em 1D. 44
Figura 26 Diferença entre modelos de Resta e SAN 47
Figura 27 Variação do tempo de pausa. 47
Figura 28 Variação da velocidade do nodo. 48
Figura 29 Modelo SAN para Random Waypoint em 2D. 49
Figura 30 Exemplo da estratégia de escolha de caminho. 50
Figura 31 Variação do tempo de pausa. 51
Figura 32 Variação da velocidade. 52
Figura 33 Efeito da escolha da estratégia para escolha de caminho. 53

Figura 34 Regras de bordas. O nodo alcança o limite da área de movimentação, deve tomar uma ação. 57
Figura 35 Modelo SAN para Random Direction, modelo de Royer, sem considerar pausa. 58
Figura 36 Autômato Dir', inclui a pausa no modelo de Royer. 59
Figura 37 Modelo SAN para Random Direction com regra de borda Bounce. . . . 60
Figura 38 Modelo SAN para Random Direction com regra de borda Wrap Around. 61
Figura 39 Modelo SAN para Random Direction com regra de borda Delete and Replace. 61
Figura 40 Distribuição espacial de nodos para padrão de Royer sem pausa. 62
Figura 41 Distribuição espacial de nodos para padrão de Royer com a pausa. Velocidade de $4m/s$ e $16m/s$ com variação da pausa. 63
Figura 42 Distribuição espacial de nodos para padrão de Royer com a pausa. Pausa de $4s$ e $64s$ com variação da velocidade. 63
Figura 43 Random Direction com regra de borda *bounce*. 64
Figura 44 Random Direction com regra de borda *Wrap Around* 64
Figura 45 *Random Direction* com regra de borda *Delete and Replace*. Variação de Velocidade e Pausa. 65
Figura 46 *Random Direction* com regra de borda *Delete and Replace*. Variação de Velocidade e Pausa. 65
Figura 47 *Random Direction* com regra de borda *Delete and Replace*. Velocidade de $4m/s$, pausa de $4s$ e variação do tamanho do movimento. 65
Figura 48 Variação do tamanho do movimento para todas combinações de velocidade (1, 4, 16, 64 metros por segundos) e tempo de pausa (“0,001”, 16, 64 e 256 segundos). 66
Figura 49 Composição da distribuição espacial a partir da pausa e movimento. . . 66
Figura 50 Autômato Dir', autômato Dir simplificado, sem o estado P, desconsiderando pausa entre movimentos. 67
Figura 51 Autômatos $LocX$ e $LocY$, representam a área de movimentação em 2D. 69
Figura 52 Autômato LocX e LocY com efeito de borda a oeste. Modelo parcial para Random Direction com *Delete and Replace* em área 2D. 69
Figura 53 Autômatos LocX e LocY com efeito de borda a oeste e a leste. Modelo parcial para Random Direction com *Delete and Replace* em área 2D. . . 70
Figura 54 Autômatos LocX e LocY completos para o padrão Random Direction com *Delete and Replace* em área 2D. 70
Figura 55 Autômato $Degree$, modela a escolha de direção do movimento. 71
Figura 56 O movimento a ser executado é a hipotenusa de um triângulo retângulo, formado pelos movimentos dos eixos oeste/leste e norte/sul. 72
Figura 57 Identificar catetos oposto e adjacente, pelo quadrante, para possibilitar a execução do movimento. 72
Figura 58 Random Direction com *Delete and Replace* em área 2D 73
Figura 59 Distribuição espacial com a variação do tamanho do movimento em: 100m, 500m, 1000m, 5000m 74
Figura 60 Variação do número de estados do autômato Degree: 4, 8, 12, 16 e 20 estados (graus). 75
Figura 61 Impacto do aumento do número de graus nos resultados obtidos. 76

Lista de Tabelas

Tabela 1 Comparação entre modelo sem pausa e modelo com pausa (0,1 s e 0,01 s). 68

Lista de Siglas

SAN	Stochastic Automata Networks	23
PEPS	Performance Evaluation Of Parallel Systems	24
RWP	Random Waypoint	30
RD	Random Direction	31
RPGM	Reference Point Group Mobility	40
DSR	Dynamic Source Routing	43
NS2	Network Simulator	43
GloMoSim	Global Mobile Information Systems Simulations Library	43

Sumário

1 Introdução 19
1.1 Objetivo do Trabalho 19
1.2 Método e Estrutura de Pesquisa 20
1.3 Estrutura do Trabalho 20

2 Redes de Autômatos Estocásticos 23
2.1 Transições de Estado 24
2.1.1 Taxas Funcionais 24
2.2 Função de Alcançabilidade 24
2.3 Exemplo de Descrição Textual de uma SAN 25

3 Padrões de Mobilidade para Redes Ad Hoc 27
3.1 Padrões de Mobilidade de Entidade 27
3.1.1 Random Walk 28
3.1.2 Versão Probabilística do Random Walk 29
3.1.3 Random Waypoint 30
3.1.4 Random Direction 31
3.1.5 Padrão de Mobilidade com Área de Movimentação sem Fronteiras 33
3.1.6 Gauss-Markov 33
3.1.7 City Section 35
3.1.8 Freeway 36
3.2 Padrões de Mobilidade de Grupo 37
3.2.1 Movimentação em Colunas 37
3.2.2 Comunidades Nômades 39
3.2.3 Movimentação em Perseguição 39
3.2.4 Movimentação por Ponto de Referência 40
3.2.5 Vetor de Movimentação 41

4 Padrão Random Waypoint 43
4.1 Modelo SAN para Random Waypoint em 1D 44
4.1.1 Analisando o Modelo 1D 45
4.2 Modelo SAN para Random Waypoint em 2D 48
4.2.1 Estratégias de Escolha de Caminhos 49
4.2.2 Analisando o Modelo 2D 52

5 Padrão Random Direction 55
5.1 Regras de Bordas 56
5.2 Modelo SAN Random Direction em 1D 57
5.2.1 SAN para Padrão de Royer 57

5.2.2 SAN para regra de borda Bounce 59
5.2.3 SAN para regra de borda Wrap Around 60
5.2.4 SAN para regra de borda Delete and Replace 61
5.2.5 Resultados para modelo em 1D 62
5.3 Modelo SAN para Random Direction em 2D 67
5.3.1 Velocidade e Direção para Random Direction em 2D 70
5.3.2 Resultados para Modelo em 2D 73

6 Conclusões e Trabalhos Futuros 77

Referências 79

Apêndice A – SAN para RWP em 1D 81

Apêndice B – SAN para RWP em 2D 85

Apêndice C – SAN para padrão RD de Royer em 1D . . 95

Apêndice D – SAN para RD - Delete and Replace em 1D 97

Apêndice E – SAN para RD - Delete and Replace em 2D 101

1 Introdução

Uma das áreas que vem se desenvolvendo muito atualmente, tanto na comunidade acadêmica quanto comercialmente, é a área de redes sem fio, que permitem os usuários acessarem serviços, independentemente de sua localização e possibilitando que se movimentem mantendo o acesso. Desta forma, usuários podem conectar seus dispositivos à rede em lugares como hotéis, aeroportos, praças ou shopping; ou ainda, pode-se considerar um dispositivo agregado a um veículo, que está constantemente em movimento e necessita estar conectado a uma rede.

Rede *Ad Hoc* é uma classe emergente de arquitetura de redes sem fio, sendo caracterizada por topologia dinâmica (variação de enlaces), limitados recursos (capacidade de transmissão e energia) e não utilização de uma estrutura fixada (*Access Point*). Essas redes destacam-se, principalmente, pela característica de não depender de uma estrutura pré-fixada.

A topologia dinâmica, devido à movimentação dos dispositivos de redes sem fio, inseriu novas características às redes de computadores, para as quais os protocolos e padrões existentes não estavam preparados, gerando muitos desafios aos pesquisadores desta área e motivam trabalhos de análise dos efeitos da mobilidade e metodologias para avaliá-los. Esta necessidade de estudo é a motivação deste trabalho, que vai tratar de análise de padrões de movimentação utilizados para avaliação de desempenho de Redes *Ad Hoc*.

1.1 Objetivo do Trabalho

Este trabalho vem ao encontro de um grande número de pesquisas de análise de desempenho de redes *Ad Hoc*, mais especificamente, na utilização de padrões de movimentação de dispositivos. O objetivo principal deste trabalho é analisar os padrões de movimentação aplicados para redes *Ad Hoc*. Para possibilitar essa análise é utilizado o formalismo markoviano de Redes de Autômatos Estocásticos. A utilização de SAN é por ser um método analítico flexível, permitindo a representação de vários aspectos dos padrões de movimentação em bom nível de detalhe, como será abordado ao longo do texto.

Complementando o objetivo geral, listam-se os objetivos específicos que guiam este trabalho:

- Estudo de padrões de movimentação utilizados na avaliação de desempenho de Redes *Ad Hoc*;
- Estudo de métodos analíticos que possam ser aplicados à avaliação de desempenho de

redes *Ad Hoc*;

- Propor modelos para análise da distribuição espacial de nodos para os padrões de movimentação *Random Direction* e *Random Waypoint*;
- Redigir relato dos resultados obtidos através dos modelos propostos, assim como comparação com outros resultados encontrados na literatura.

1.2 Método e Estrutura de Pesquisa

A Figura 1 representa as principais atividades desenvolvidas na condução desta pesquisa. Inicialmente, identificou-se a área de pesquisa a ser estudada. Logo após, foi feito levantamento na literatura das principais métricas de avaliação de desempenho de Redes *Ad Hoc*, bem como a identificação dos métodos utilizados para estas análises.

Com a revisão bibliográfica, foi possível definir os objetivos do trabalho. O objetivo principal foi de apresentar propostas de modelos analíticos, utilizando Redes de Autômatos Estocásticos, para análise da distribuição espacial de nodos. A utilização de um formalismo markoviano estruturado, Redes de Autômatos Estocásticos, possibilitou que a partir de um primeiro modelo proposto, fossem desenvolvidos outros modelos, fazendo extensões do modelo original.

As validações dos modelos foram efetuadas, comparando-se os resultados obtidos com outros resultados encontrados na literatura. Para alguns modelos, foi possível efetuar uma comparação numérica de resultados, possibilitando uma análise mais acurada dos modelos propostos. Porém, em alguns casos, foi possível somente avaliar visualmente os gráficos de comportamento dos modelos. Por fim, foi redigido o artigo [7] e esta dissertação relatando todo o trabalho executado, inclusive os resultados obtidos.

1.3 Estrutura do Trabalho

Este trabalho está dividido em seis capítulos, sendo esta Introdução o primeiro. No Capítulo 2 é feita uma breve descrição de Redes de Autômatos Estocásticos, que é um formalismo markoviano estruturado e será o método utilizado para a modelagem do comportamento dos padrões de movimentação.

O Capítulo 3 traz um estudo sobre os padrões de movimentação utilizados em análises de desempenho de Redes *Ad Hoc*. São descritas as principais características de comportamento de nodos para diversos padrões de movimentação. Este capítulo foi baseado no *survey* publicado por Tracy Camp, Jeff Boleng e Vanessa Davies [6].

Os Capítulos 4 e 5 descrevem, em maior nível de detalhes, os comportamentos dos nodos nos padrões de movimentação *Random Waypoint* e *Random Direction*, respectivamente.

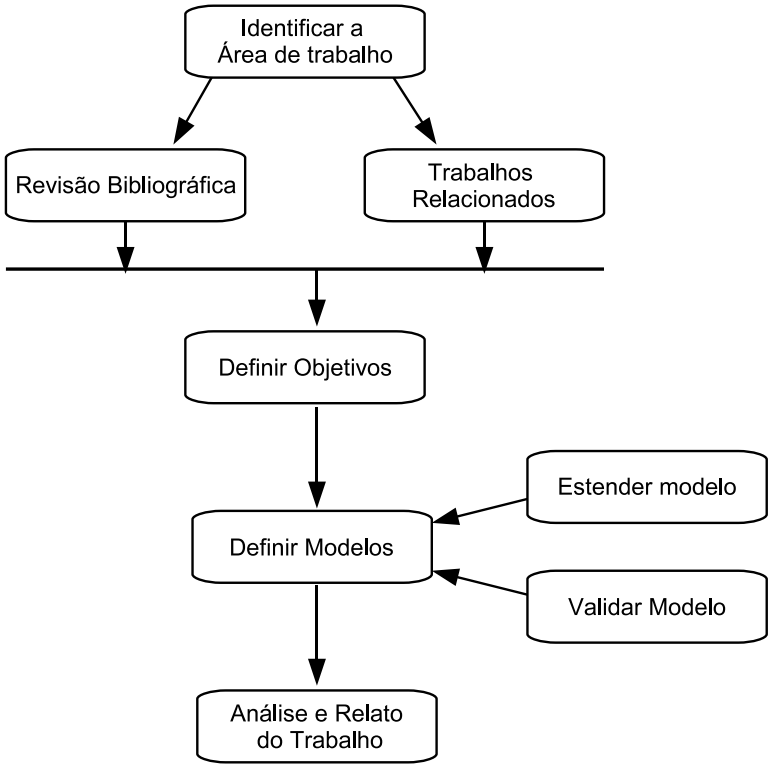


Figura 1 – Atividades desenvolvidas na condução da pesquisa

Em cada um destes capítulos são apresentados modelos, utilizando Redes de Autômatos Estocásticos, para analisar a distribuição espacial de nodos. Nestes Capítulos, os dois padrões de movimentação são analisados detalhadamente quanto às possíveis variações (velocidades, escolhas de caminhos e paradas após execução de movimentos) e seu impacto sobre a distribuição espacial de nodos. O Capítulo 6 discorre sobre as conclusões e trabalhos futuros.

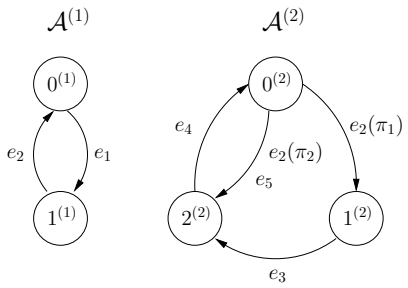
2 Redes de Autômatos Estocásticos

O formalismo Redes de Autômatos Estocásticos (*Stochastic Automata Networks* - SAN) foi proposto por Plateau [17]. A idéia básica de SAN é descrever um sistema complexo através de uma coleção de subsistemas com comportamentos independentes (eventos locais) e, ocasionalmente, interdependentes (taxas funcionais e eventos sincronizantes). Cada subsistema é descrito como um autômato estocástico, um autômato com eventos que são setados com informações de probabilidade e tempo. Para um modelo SAN sempre existe uma cadeia de Markov equivalente [5, 20], porém, SAN diminui o problema da explosão do espaço de estados.

Autômato é um modelo matemático de um sistema, com entradas e saídas discretas. O sistema pode se encontrar em qualquer estado dentro de um número finito de estados que compõem este sistema [17]. Um autômato é um conjunto de estados, que representam uma realidade, e as possíveis transições de estados, que representam as mudanças de estado desta realidade.

O estado de um autômato resume toda a informação referente a seu passado, pois as entradas passadas são necessárias para determinar o estado atual. Para um determinado conjunto de estados, um sistema poderá assumir somente um estado a cada momento, e este estado irá variar de acordo com a nova entrada recebida [17].

O estado individual de cada autômato é chamado de estado local. O estado global de uma SAN é definido como a combinação de todos os estados locais de cada autômato componente desta.



$$f = \left[\left(st\ \mathcal{A}^{(2)} == 0^{(2)}\right) * \lambda\right] + \left[\left(st\ \mathcal{A}^{(2)} == 2^{(2)}\right) * \gamma\right]$$

Type	Event	Rate
loc	e_1	f
syn	e_2	μ
loc	e_3	σ
loc	e_4	δ
loc	e_5	τ

Figura 2 – Exemplo de um modelo SAN.

2.1 Transições de Estado

As transições de estado do sistema ocorrem através de eventos. Há dois tipos de eventos que modificam o estado de um modelo SAN: *eventos locais* e *eventos sincronizantes*. Eventos locais alteram o estado de um único autômato. Eventos sincronizantes modificam estados de mais de um autômato simultaneamente, possibilitando a interação entre vários autômatos.

2.1.1 Taxas Funcionais

Além dos eventos sincronizantes, taxas funcionais também podem ser utilizadas para representar interações entre autômatos. Ambos os tipos de eventos (locais e sincronizantes) podem conter taxas funcionais. Essas taxas são definidas por funções que são avaliadas conforme os estados atuais do modelo SAN. Desta forma, pode-se setar um evento de forma que sua taxa de ocorrência varie de acordo com o estado de um outro, ou de vários, autômato(s) do sistema.

A Figura 2 apresenta um modelo SAN com dois autômatos ($A^{(1)}$ e $A^{(2)}$), com um evento sincronizante (e_2) e quatro eventos locais (e_1, e_3, e_4 e e_5). Neste exemplo, a taxa do evento e_1 não é uma constante, mas uma taxa funcional f descrita conforme notação SAN[1] para a ferramenta *Performance Evaluation Of Parallel Systems* - PEPS [2]. A interpretação de f define a possibilidade de transição do estado $0^{(1)}$ para $1^{(1)}$ com a taxa λ se o autômato $\mathcal{A}^{(2)}$ está no estado $0^{(2)}$, ou a taxa γ se autômato $\mathcal{A}^{(2)}$ está no estado $2^{(2)}$. Se o autômato $\mathcal{A}^{(2)}$ está no estado $1^{(2)}$, a transição do estado $0^{(1)}$ para $1^{(1)}$ não ocorre (taxa é igual a zero). É importante observar que, o uso de funções permite uma forma compacta e flexível para descrever um comportamento alternativo para os eventos (local ou sincronizante).

2.2 Função de Alcançabilidade

A Figura 3 mostra a cadeia de Markov equivalente para o modelo SAN apresentado (fig. 2). Assumindo o estado $0^{(1)}0^{(2)}$ como um estado inicial, somente cinco de seis estados que estão na cadeia de markov são alcançáveis. Para permitir redução do modelo, é necessário indicar os estados globais alcançáveis do modelo SAN através de uma função de *reachability*. Para alguns modelos SAN, a definição da função de alcançabilidade não é muito simples. De fato, as experiências com SAN demonstram que isto é um aspecto negativo deste formalismo. Contudo, o conjunto de estados alcançáveis pode ser encontrado, computacionalmente, a partir de um dado estado alcançável.

[1] O interpretador de função pode ser visto com o avaliador de expressões de uma linguagem não tipada, exemplo linguagem C. Cada comparação é avaliada como valor 1 (verdadeiro) ou valor 0 (falso).

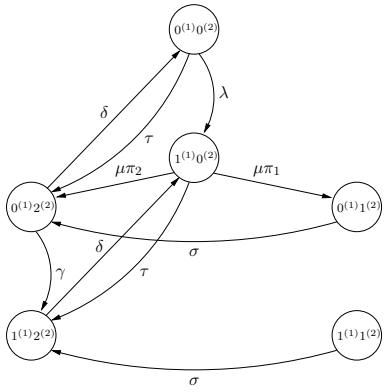


Figura 3 – Cadeia de Markov Equivalente.

Para o modelo na Figura 2, a função de alcançabilidade exclui o estado global $1^{(1)}1^{(2)}$, sendo:

$$Reachability = !\left[\left(st\ \mathcal{A}^{(1)} == 1^{(1)}\right)\ \&\&\ \left(st\ \mathcal{A}^{(2)} == 1^{(2)}\right)\right]$$

2.3 Exemplo de Descrição Textual de uma SAN

A descrição textual da SAN, apresentada na Figura 2, é demonstrada a seguir. Esta descrição pode ser utilizada como entrada para avaliação na ferramenta PEPS [2].

```
identifiers

    lambda = 1;
    gamma  = 2;
    mu     = 3;
    sigma  = 4;
    delta  = 5;
    tau    = 6;
    pi1    = 0.3;
    pi2    = 1 - pi1;
    efe = ( ((st A2 == N0)*lambda)+((st A2 == N2)*gamma) );

events

    loc e1 (efe)
    syn e2 (mu)
    loc e3 (sigma)
    loc e4 (delta)
    loc e5 (tau)
```

```
reachability = !( (st A1 == N1) && (st A2 == N1) );

network exemplo(continuous)

    aut A1
        stt N0
            to(N1) e1
        stt N1
            to(N0) e2

    aut A2
        stt N0
            to(N1) e2(pi1)
            to(N2) e2(pi2) e5
        stt N1
            to(N2) e3
        stt N2
            to(N0) e4

results
    prob_A1_N0 = (st A1 == N0);
    prob_A2_N2 = (st A2 == N2);
    prob_A1_Menor_A2 = (st A1 < st A2);
```

3 Padrões de Mobilidade para Redes Ad Hoc

O grande desafio na avaliação de redes *Ad Hoc* é, certamente, a movimentação dos dispositivos que fazem parte da rede. Para possibilitar análise de desempenho de redes *Ad Hoc* é necessário utilizar algum padrão de movimentação. Uma possibilidade é a utilização de *trace*, que é o padrão obtido através da observação de um sistema real. Porém, obter o trace pode ser muito difícil, devido aos elevados número de participantes e tempo de observação necessários para obter um padrão consistente. A outra possibilidade é a utilização de padrões sintéticos, os quais definem comportamentos da movimentação para os dispositivos [6].

Neste capítulo serão apresentados alguns padrões sintéticos de movimentação[1], dos quais onze dos treze padrões foram apresentados no *survey* da Tracy Camp, Jeff Boleng e Vanessa Davies [6]. Existem duas grandes classes de padrões de mobilidade: mobilidade de entidade, onde um nodo se movimenta independentemente dos outros, e mobilidade de grupo, onde a movimentação de um nodo depende de outros nodos.

3.1 Padrões de Mobilidade de Entidade

Os padrões de mobilidade de entidade representam a movimentação dos nodos de forma independente, ou seja, não existem relações nas decisões de movimentos de nodos diferentes.

Aqui são citados oito padrões de mobilidade de entidade, que serão descritos nas seções subseqüentes:

- *Random Walk*
- Versão Probabilística do *Random Walk*
- *Random Waypoint*
- *Random Direction*
- Padrão de Mobilidade com Área de Movimentação sem Fronteiras
- *Gauss-Markov*
- *City Section*

[1]Neste trabalho, serão referenciado os padrões sintéticos de movimentação apenas como padrões de movimentação ou padrão de mobilidade, a fim de simplificar a redação.

- *Freeway*

3.1.1 Random Walk

Random Walk é um padrão simples de mobilidade, que se baseia em escolhas aleatórias de direções e de velocidades. Neste padrão, o nodo móvel movimenta-se de uma posição inicial para uma nova posição, variando de forma aleatória a direção e velocidade, dentro de variações de valores pré-estabelecidas, gerando um comportamento como demonstrado no gráfico de movimentação na Figura 4, apresentada em [6].

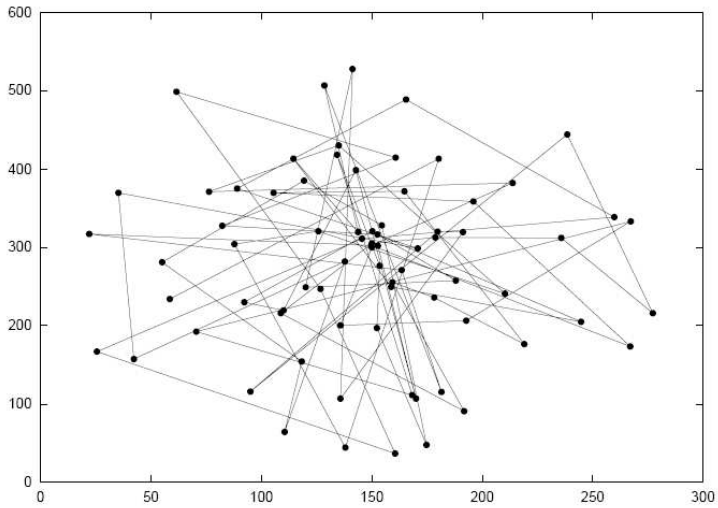


Figura 4 – Exemplo de movimentação gerada no padrão Random Walk.

Cada movimento ocorre em um determinado intervalo de tempo t ou em uma distância constante d. Percorrida a distância d ou passado o tempo t, o nodo móvel calcula nova direção e velocidade para o novo deslocamento. Caso um deslocamento alcance os limites da área de movimentação, uma nova direção será calculada de modo a manter o nodo dentro dos limites da área.

Um exemplo de movimentação de um nodo, de acordo com este padrão, é apresentado na Figura 5. Neste exemplo, o movimento do nodo é determinado por um tempo t, ou seja, ao decidir por um movimento, ele o executa por t unidades de tempo. O nodo inicia o movimento no ponto A, após uma decisão sobre o movimento inicial o nodo e movimenta-se até o ponto B. Neste ponto, o nodo decidirá por uma nova velocidade v e ângulo α, esses valores são escolhidos de forma aleatória. Após definido v e α, o nodo efetua o movimento 2 no tempo t. Passado t ou alcançada a borda da área considerada, o nodo deve decidir pelo próximo movimento.

A definição dos valores de d ou t determinará o padrão do movimento dentro da área de movimentação. Valores pequenos para estas variáveis resultam em um padrão de movimento em que nodo manter-se-á em uma pequena região da área de movimentação, onde haverá pouca

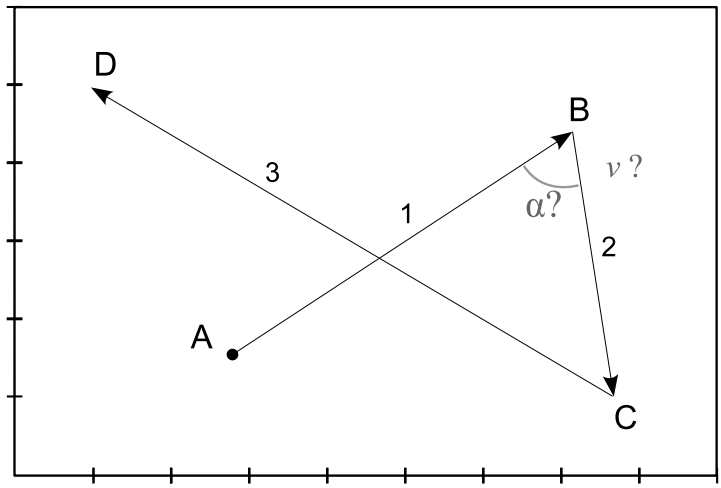


Figura 5 – Movimentação no Random Walk.

movimentação em relação à área de movimentação, como pode ser visto na Figura 6, apresentada em [6]. Escolher valores maiores para estas variáveis fará com que o nodo se espalhe mais pela área de movimentação.

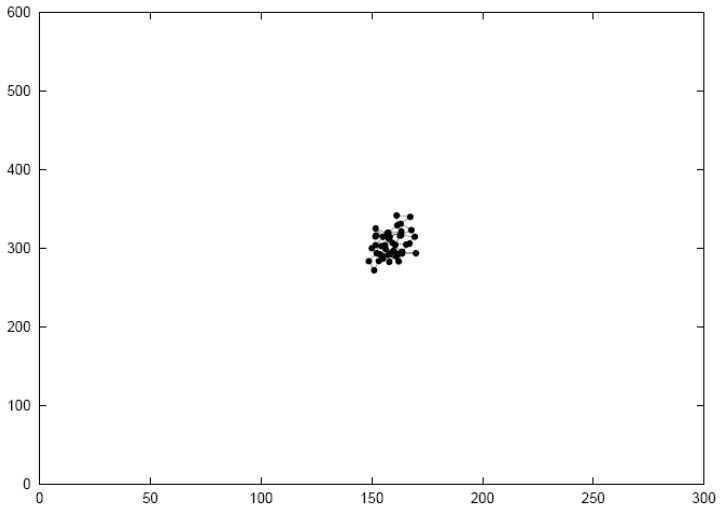


Figura 6 – Exemplo de movimentação gerada pelo padrão Random Walk, usando baixos valores para distância ou tempo de movimentação.

3.1.2 Versão Probabilística do Random Walk

Como o próprio nome indica, este padrão é baseado no padrão *Random Walk*, aplicando probabilidades nas decisões de cada novo movimento, tendo uma velocidade pré-definida para a simulação. É informada, no início da simulação, uma matriz de probabilidades e a cada decisão de uma nova direção para o nodo é aplicada esta matriz. Assim a movimentação do nodo será mais comportada, evitando mudanças bruscas de direção e sentido, como ocorrem na versão

original do padrão. Este comportamento pode ser visto na Figura 7, apresentada em [6]. A matriz de probabilidades é configurada de forma que tenha maior chance nodo manter a mesma direção a cada escolha de novo movimento.

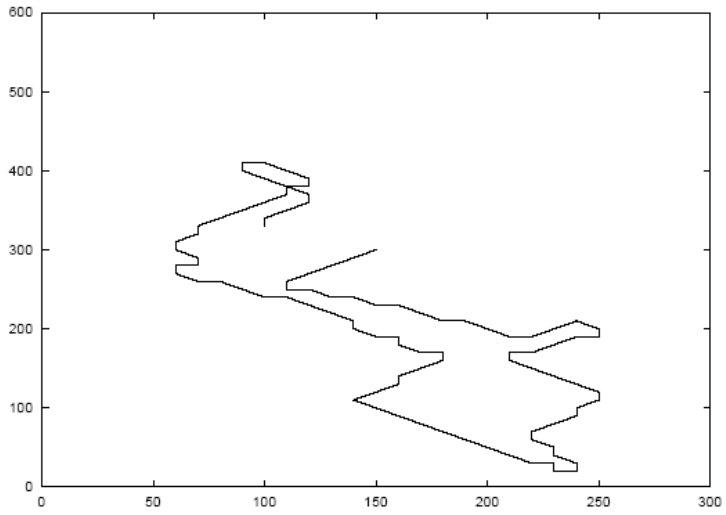


Figura 7 – Exemplo de movimentação gerada pela versão probabilística do Random Walk.

3.1.3 Random Waypoint

O padrão de movimentação *Random Waypoint* é um dos padrões mais utilizado em simulações para análise de desempenho de redes *Ad Hoc*. Este padrão é caracterizado pela escolha aleatória de destino.

Como pode ser visto no exemplo da Figura 8, assumindo A como ponto inicial, o nodo móvel escolhe aleatoriamente um ponto de destino dentro da área de movimentação (B) e define aleatoriamente a velocidade (v) dentro de um variação pré-definida e executa o movimento até que alcance o ponto B. Ao chegar em B, ele ficará parado por um período de tempo, chamado de pausa. Após este período, ele inicia novamente a operação de movimento, selecionando um novo ponto de destino (C) e nova velocidade de deslocamento. Na Figura 9, apresentada em [6], é demonstrado um exemplo de caminhos gerados pela movimentação do nodo.

Em muitas avaliações de desempenho onde é empregado este padrão, a distribuição inicial dos nodos é feita de forma aleatória. Isto acaba gerando, no período inicial de experimento, uma grande variação no número de vizinhos dos nodos, como demonstrado no gráfico da Figura 10, apresentada em [6]. Essa grande variação impacta nos resultados de avaliação de desempenho destas redes, pois o desempenho de uma rede, na maioria dos casos, tem relação com o número de vizinhos dos nodos. Algumas maneiras de evitar estes problemas, segundo [6]:

- Sempre iniciar a simulação da mesma forma;

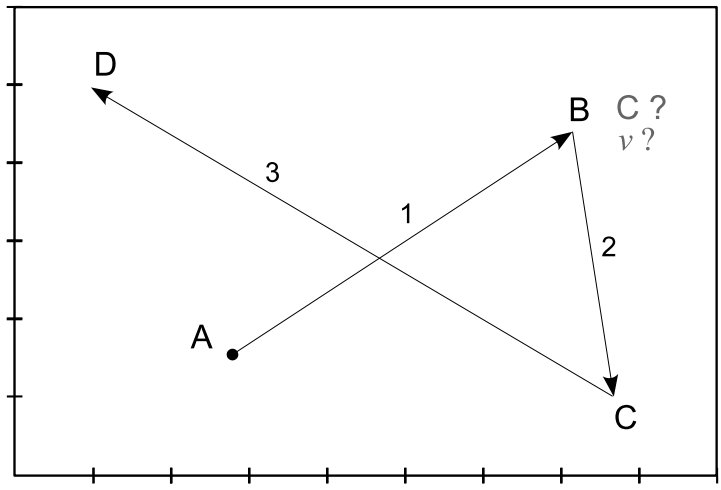


Figura 8 – Movimentação no Random Waypoint.

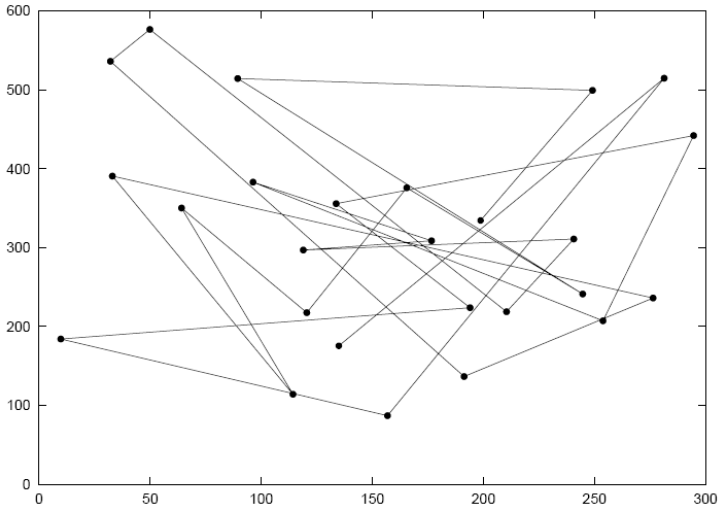


Figura 9 – Exemplo de movimentação gerada pelo padrão Random Waypoint.

- Armazenar mapas de padrões de inicialização para as simulações;
- Descartar os 1000 segundos iniciais de simulação.

Outro fator muito importante neste padrão é a escolha dos valores para velocidade e tempo de parada. Nodos móveis mais rápidos e com tempos de parada mais longos, geram cenários mais estáveis, comparados a nodos mais lentos e com curtas paradas. Ou seja, quanto menor a movimentação do nodo, mais estável será o cenário obtido.

3.1.4 Random Direction

O padrão *Random Direction* (RD) surgiu com a finalidade de minimizar o problema da variação de vizinhos que ocorre no padrão de *Random Waypoint*. Diferencialmente ao RWP, este

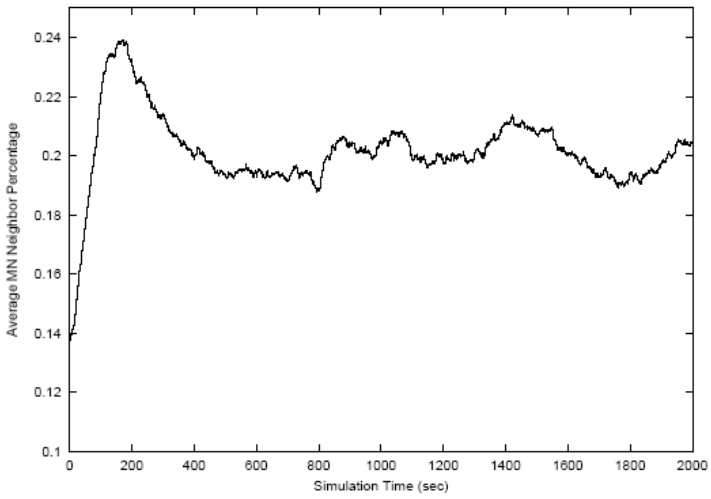


Figura 10 – Variação do número de vizinhos no Random Waypoint.

padrão de movimentação é baseado na escolha aleatória de direção e sentido para o movimento. Existem algumas variações para o seu comportamento, como por exemplo: a definição do comportamento do nodo ao incidir sobre os limites da área de movimentação (borda) e a definição do limitador de movimentos. Estas variações estão descritas em detalhes no Capítulo 5.

Na Figura 11, apresentada em [6], é demonstrado um exemplo de caminhos percorridos por um nodo no padrão RD, considerando que os movimentos são executados até o limite da área de movimentação (o limitador do movimento é a borda).

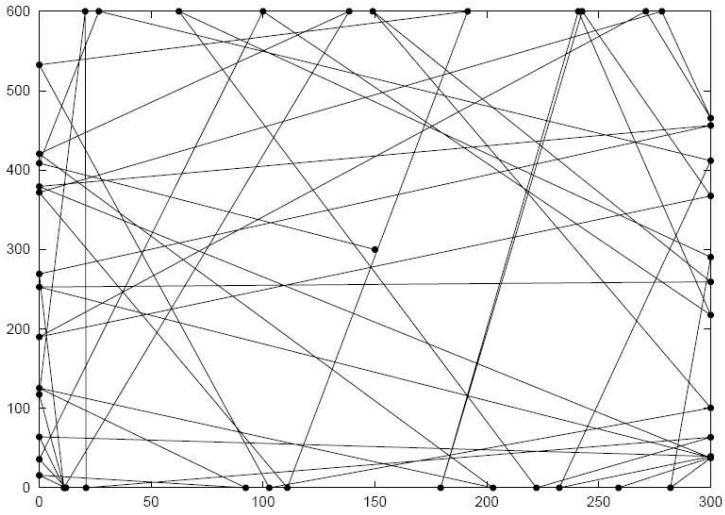


Figura 11 – Exemplo de movimentação gerada pelo padrão Random Direction.

No exemplo, demonstrado na Figura 12, o nodo móvel inicia em um ponto qualquer, no caso A, escolhe uma direção para o movimento e executa-o (movimento 1) até que alcance o limite da área de movimentação (ponto B). Em B, o nodo pode executar uma pausa e, depois, inicia novo processo de definição de movimento, escolhendo nova direção e sentido, de forma a manter-se dentro da área de movimentação.

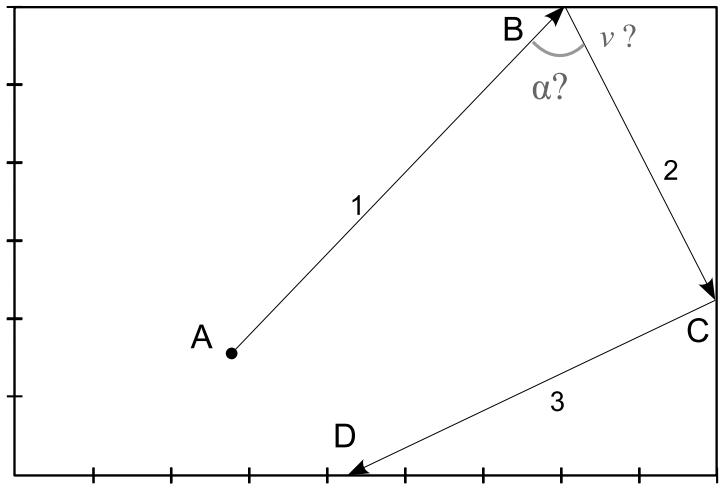


Figura 12 – Movimentação no Random Direction.

3.1.5 Padrão de Mobilidade com Área de Movimentação sem Fronteiras

Os padrões demonstrados até agora apresentam característica de *Memoryless*, ou seja, um novo movimento do nodo não depende dos movimentos anteriores. O Padrão de Mobilidade com Área de Movimentação sem Fronteiras, proposto em [8], mantém uma relação entre um novo movimento e os anteriores. Cada nova combinação de velocidade e direção é gerada a partir da combinação anterior, gerando um padrão de movimentação como demonstrado na Figura 13, apresentada em [6].

O movimento de cada nodo móvel na área de movimentação é determinado por dois elementos em $\overrightarrow{v} = (v, \theta)$, onde v é a velocidade e θ é a direção, os quais são atualizados a cada período de tempo Δt. A definição de novas velocidade e direção é obtida através de um conjunto de equações que pode ser visto em [6].

Outra característica deste padrão é quanto à área de movimentação. Nos outros padrões, se o nodo móvel atinge uma borda de limite da área de movimentação, este inicia um novo movimento, com nova direção, de modo a manter-se dentro desta área, como um efeito de reflexão. Neste padrão, o nodo ao atingir uma borda é transferido para o outro lado da área. Desta forma, seria como se a borda esquerda fosse ligada à borda direita, assim como a superior com a inferior, em uma imaginável área retangular, como pode ser visto na Figura 14.

3.1.6 Gauss-Markov

O padrão de mobilidade *Gauss-Markov* foi originalmente proposto para simulação de redes de celulares [14], porém também foi aplicado em simulação de redes *Ad Hoc* [6].

Um movimento, de acordo com este padrão, é definido pelo conjunto $mov(n, v, d)$, sendo

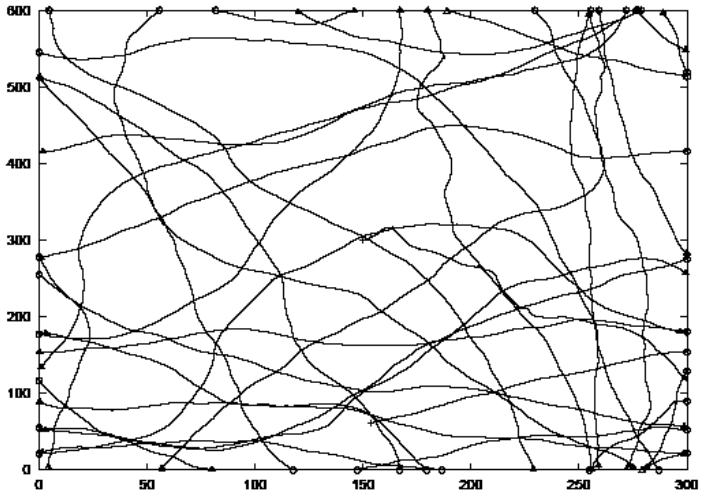


Figura 13 – Exemplo de movimentação gerada pelo Padrão de Mobilidade com Área de Movimentação sem Fronteiras.

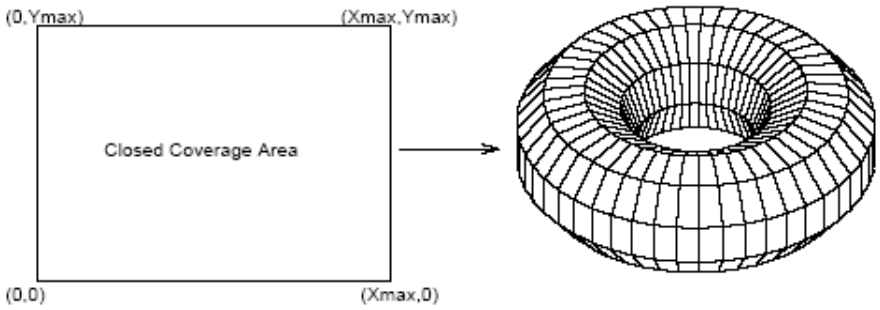


Figura 14 – Área de movimentação redesenhada para uma área sem fronteiras.

n um período de tempo, v velocidade e d direção. O nodo inicia em um determinado ponto A (fig.15), movimentando-se em uma determinada direção, indicada pela seta pontilhada 1. No tempo percorrido no momento n, o nodo chega ao ponto B, onde deve definir nova velocidade e direção, a partir dos valores do movimento anterior. A nova direção, indicada pela seta pontilhada 2, e velocidade são obtidas a partir da direção e velocidade do movimento entre A e B. A relação entre novo movimento e movimento anterior é definida por equações que podem ser encontradas em [6].

Para evitar que o nodo saia da área de movimentação, quando este atinge as regiões próximas as bordas (partes marcadas em cinza, delimitadas pelas linhas pontilhadas indicadas na Figura 15), a variável d (direção) do novo movimento do nodo será setada com valores das constantes m_x (são as setas m_1, m_2, ..., m_8), conforme a região em que o nodo estiver localizado. Desta forma, o nodo deverá retornar em direção ao centro da área de movimentação.

Na Figura 16 é apresentado um exemplo de caminhos gerados por um nodo seguindo o padrão de movimentação *Gauss-Markov*.

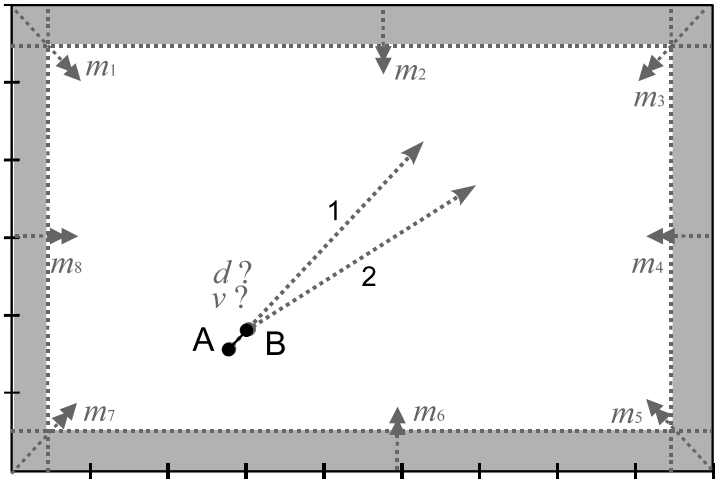


Figura 15 – Movimentação no Gauss-Markov.

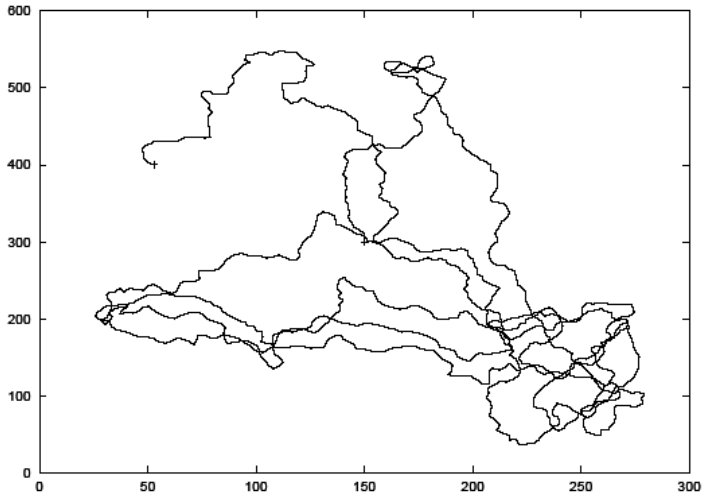


Figura 16 – Exemplo de movimentação gerada no Gauss-Markov.

3.1.7 City Section

A idéia básica do *City Section* é modelar a movimentação de nodos pelas ruas de uma cidade. A área de movimentação é uma grade, com caminhos (ruas) por onde os nodos poderão transitar e blocos (quarteirões ou quadras), que devem ser desviados (os nodos movimentam-se pelas ruas entre os blocos). Os sentidos e velocidades das ruas poderão ser modelados neste padrão. Desta forma, as decisões de caminhos devem considerar estas regras (sentido e velocidade permitidos), de forma que os nodos escolham os melhores caminhos para chegarem aos seus destinos.

A movimentação é definida de acordo com os pontos de origem e de destino. Para um determinado par de pontos (origem e destino), é selecionado o melhor caminho a ser percorrido para completar o movimento. Como pode ser visto na Figura 17 (apresentada em [6]), o nodo

inicia no ponto $(1, 1)$. Define que o destino é $(5, 4)$, então segue até o destino pelo caminho menos custoso, de acordo com as limitações de cada trecho. Neste movimento o nodo segue o caminho: $(1, 1) \rightarrow (3, 1) \rightarrow (3, 4) \rightarrow (5, 4)$.

Além das definições de sentidos e velocidades permitidas para cada trecho, paradas em determinadas intersecções e destinos (representando semáforos ou obstáculos) podem ser modeladas, de forma a possibilitar uma representação mais detalhada de área de movimentação em situações reais.

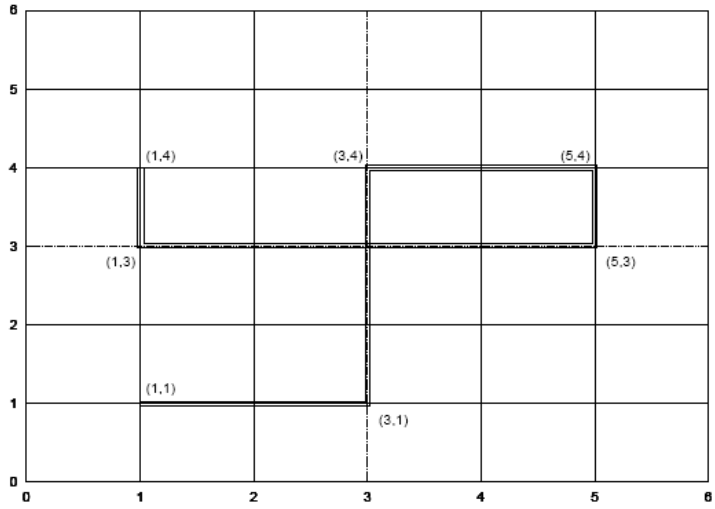


Figura 17 – Exemplo de movimentação gerada no City Section.

3.1.8 Freeway

Proposto em [1], o objetivo deste padrão é representar o comportamento de nodos móveis em estradas, ex. veículos que estão trafegando em um conjunto de rodovias. Neste padrão são criados mapas, que definem as estradas dentro da área de movimentação, por onde os nodos poderão trafegar, como pode ser visto no exemplo da Figura 18, apresentada em [1]. A velocidade do nodo pode variar a cada período de tempo, mantendo uma relação com a velocidade anterior. Caso ocorra a situação de dois nodos trafegando na mesma estrada e no mesmo sentido, o nodo que se encontra atrás deverá respeitar a velocidade do nodo logo a sua frente, de forma a não ocorrer, o que seria numa situação real, uma ultrapassagem, bem como respeitar um limite de distância do nodo da frente.

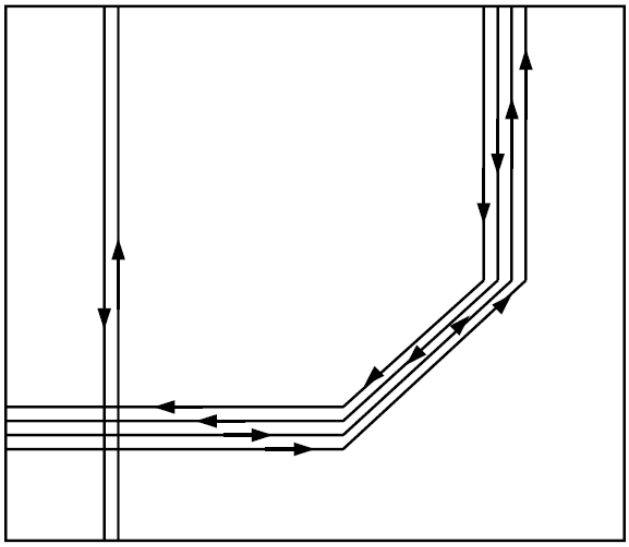


Figura 18 – Exemplo de mapa de movimentação para o padrão de Movimentação Freeway.

3.2 Padrões de Mobilidade de Grupo

Quando se deseja representar movimentação de conjuntos de nodos, os padrões de mobilidade de entidade não são eficientes, tornando-se necessária a utilização de padrões de mobilidade de grupo. Nestes padrões, grupos de nodos possuem algum comportamento em comum na decisão de um novo movimento.

Nesta seção são apresentados alguns padrões de mobilidade em grupo:

- Movimentação em Colunas
- Comunidades Nômades
- Movimento em Perseguição
- Ponto de Referência de Grupo
- Vetor de Movimentação

3.2.1 Movimentação em Colunas

O padrão de Movimentação em Colunas pode ser usado para representar nodos que andam alinhados, por exemplo: uma coluna de soldados fazendo uma busca em uma área, como a procura de minas em uma determinada região.

A movimentação é feita através de um conjunto de pontos de referência, dispostos em uma coluna ou uma linha, como pode ser visto na Figura 19, apresentada em [6]. Cada nodo (representado por circunferências) possui um ponto de referência nesta linha (indicado pelos pontos

pretos) e pode movimentar-se utilizando algum padrão de mobilidade de entidade, mantendo-se em torno do seu ponto de referência. Esta linha de pontos de referência movimenta-se em direção a um destino, podendo girar em torno de si, até um ângulo escolhido para o movimento, como pode ser visto na Figura 20.

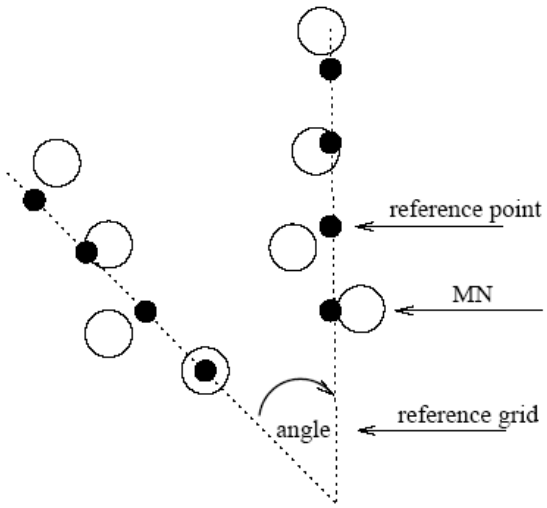


Figura 19 – Coluna de pontos de referência do padrão de movimentação em Colunas.

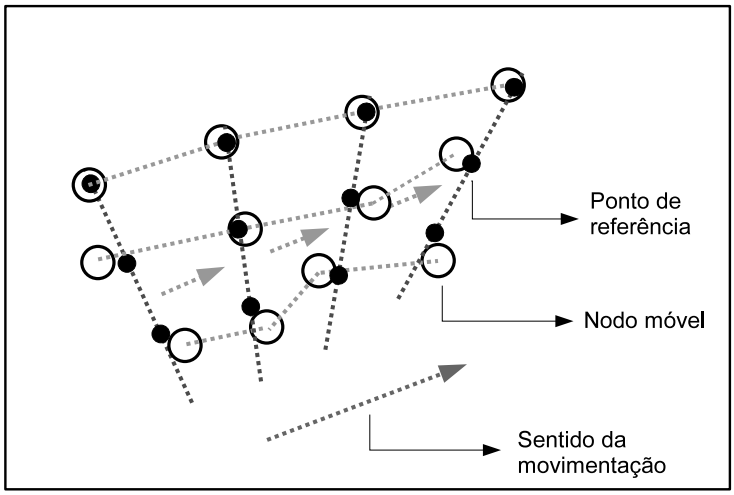


Figura 20 – Movimentação dos nodos no padrão de movimentação em colunas.

Em [6], é demonstrada a movimentação dos nodos, conforme este padrão, como pode ser visto na Figura 21, com duas variações: a primeira representada pelas linhas pontilhadas, onde é demonstrada uma movimentação perpendicular da coluna em relação ao sentido, e nas linhas contínuas a coluna está de forma paralela ao sentido.

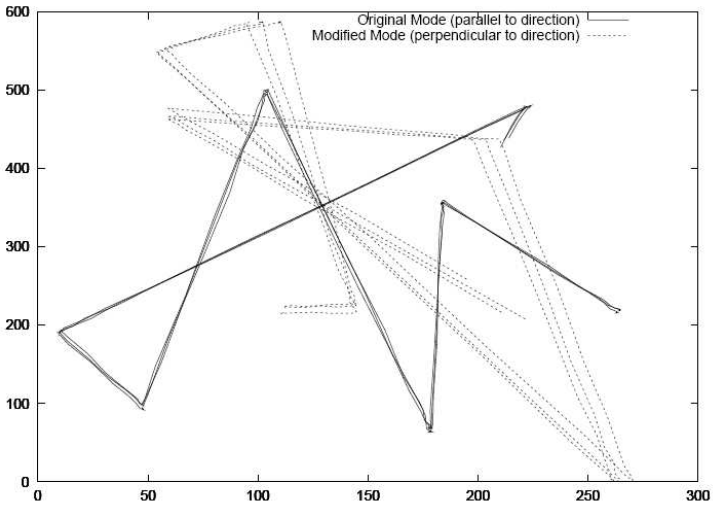


Figura 21 – Exemplo de movimentação gerado no padrão de movimentação em colunas.

3.2.2 Comunidades Nômades

No padrão de movimentação de Comunidades Nômades, grupos de nodos movimentam-se coletivamente de um ponto para outro e possuem um ponto de referência, o qual determina a posição do grupo. Cada nodo pode mover-se de forma independente, conforme um determinado padrão de movimentação de entidade, mantendo uma relação de proximidade com o ponto de referência do seu grupo.

A movimentação do grupo é determinada pela mudança do ponto de referência. Quando ocorre esta mudança, todos os nodos viajam juntos, mantendo a sua relação com o ponto de referência, podendo considerar a movimentação individual de cada nodo. Um exemplo de movimentação é apresentado na Figura 22, onde ocorre a mudança do ponto de referência (representado pelo círculo preto) de A para B. Os seis nodos movimentam-se em torno do ponto de referência, podendo efetuar movimentos independentes, como é o caso do nodo que executaria o movimento até o ponto P_1, porém com o movimento individual indicado pela seta 2, seu destino passou a ser o ponto P_2.

3.2.3 Movimentação em Perseguição

Neste padrão os nodos movimentam-se como em uma perseguição. Como mostrado na Figura 23, onde o círculo preto representa o nodo perseguido P, que se movimenta de A para B. Os outros nodos, devem efetuar o movimento de acordo com algum padrão de mobilidade aleatório, como por exemplo: *Random Walk*, porém a nova posição do nodo deve restringir-se a uma área próxima à P, representado pelo retângulo de pontilhado cinza. O movimento 3, representa o movimento de um nodo, perseguindo o nodo P.

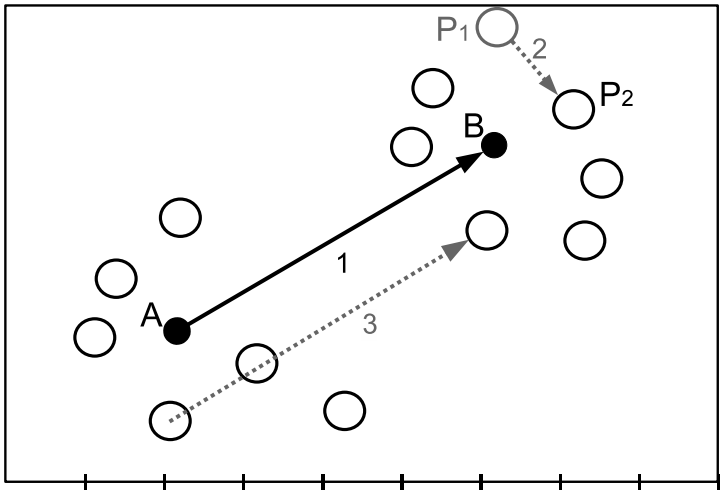


Figura 22 – Movimentação de Comunidades Nômades.

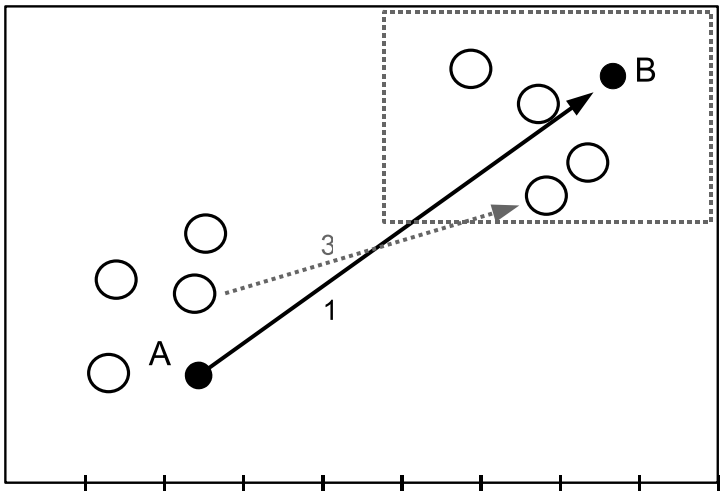


Figura 23 – Padrão de Movimentação em Perseguição.

3.2.4 Movimentação por Ponto de Referência

O padrão de movimentação por Ponto de Referência (*Reference Point Group* - RPGM), proposto em [10], é um padrão muito flexível para modelagem de movimentação em grupo. Neste, os nodos são agrupados de acordo com uma lógica ou padrão de comportamento de movimentação entre eles. Este comportamento define as variáveis: velocidade, direção, localização e aceleração. O padrão também permite movimentação dos nodos de forma aleatória e independente, além da movimentação do grupo.

Com este padrão é possível modelar variados comportamentos de movimentação de grupo, definindo diferentes padrões de lógica de movimentação dos grupos. Em [10] são descritos vários cenários onde pode ser aplicados este padrão, como: movimentação de tropas em um campo de batalha, onde cada grupo possui um objetivo; equipes de salvamento (paramédicos, policiais, bombeiros, ...) em uma área de desastre, cada equipe possui certo comportamento

e está espalhada por toda a área. Na Figura 24 é apresentado um exemplo, onde existem dois grupos diferentes (Group1 e Group2) e cada um tem seu comportamento definido. O movimento dos nodos, representado pelas setas, é definido pelo grupo ao qual o nodo faz parte.

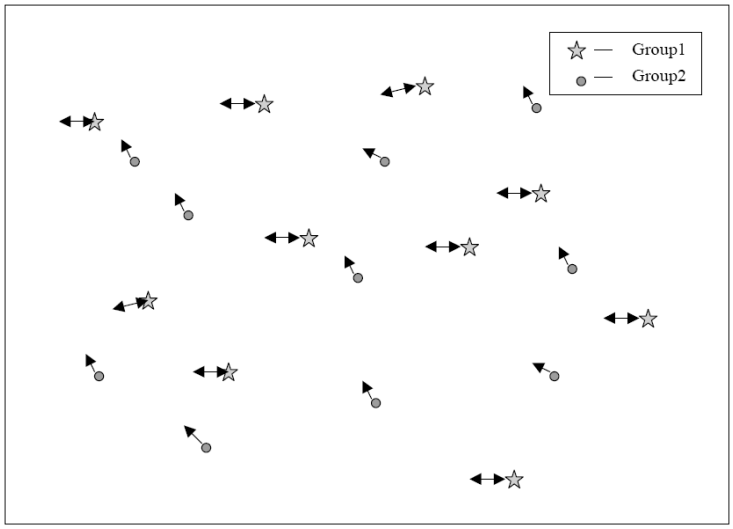


Figura 24 – Exemplo de comportamento da movimentação dos nodos de dois grupos diferentes no RPGM.

3.2.5 Vetor de Movimentação

Proposto em [13], o padrão por Vetor de Movimentação tem como objetivo criar um padrão de mobilidade de grupo mais próximo da movimentação natural dos nodos em um cenário, evitando as mudanças bruscas em movimentos, normalmente gerados por outros padrões de movimentação aleatória.

A mobilidade do nodo $\overrightarrow{M}$ é definida em um vetor (x_v, y_v), que expressa a direção do movimento e a velocidade, sendo a velocidade definida pelo período de tempo necessário para chegar ao destino (x_v, y_v).

Qualquer movimentação $\overrightarrow{M}$ pode ser definida como a soma de dois vetores: vetor Base, $\overrightarrow{B} = (bx_v, by_v)$, e vetor de Desvio, $\overrightarrow{V}(vx_v, vx_v)$. Onde, o vetor Base indica valores máximos para velocidade e direção e o vetor de desvio, expressa variação no vetor base. Então:

$\overrightarrow{M} = \overrightarrow{B} + \alpha * \overrightarrow{V}$, onde α é o fator de aceleração definido para a simulação, que pode ser variável.

Com este esquema é possível modelar muitas situações existentes em mobilidade, com algumas vantagens: simplificação da atualização de posição, fácil implementação e oportunidade de predição de movimentação. [11]

4 Padrão Random Waypoint

O padrão de mobilidade *Random Waypoint* foi inicialmente utilizado por Johnson e Maltz, em [12], na análise de desempenho do *Dynamic Source Routing* (DSR). Tal padrão está implementado nas principais ferramentas de simulação: *NS2* [22] e *GloMoSim* [21], e é vastamente utilizado em avaliações de desempenho de redes *Ad Hoc*.

De forma geral, o comportamento do nodo móvel no padrão RWP é:

1. Escolhe destino na área de movimentação;
2. move-se até o destino com velocidade constante *v*. A velocidade é escolhida no intervalo (v_{min},v_{max}] em uma distribuída uniforme;
3. ao alcançar o destino, o nodo fica parado por um determinado tempo de pausa;
4. reinicia o processo no passo 1.

Em [4], o autor apresenta um estudo analítico da distribuição espacial de nodos para o padrão de mobilidade *Random Waypoint*. Os resultados apresentados, são validados através de simulação. O RWP é modelado considerando área de movimentação em uma dimensão, como um segmento de reta, e duas dimensões, como em uma área quadrada. Como no RWP cada nodo move-se independentemente dos outros nodos, todos têm as mesmas características de movimento. Nos dois casos modelados, 1D e 2D, é considerada apenas a movimentação de um nodo sem tempo de pausa.

Resta e Santi, em [18], generalizaram a análise de [4] para considerar variações no tempo de pausa e na velocidade. Eles derivaram uma equação para a distribuição espacial de nodos considerando área de movimentação com uma e duas dimensões. Porém, no caso de duas dimensões, a equação resulta em uma aproximação.

Os estudos demonstram que, embora a posição inicial dos nodos seja uma distribuição uniforme, o padrão de movimentação altera esta distribuição durante o movimento. Este efeito ocorre por que o nodo tende a passar no centro da área de movimentação com maior freqüência. Outra característica é que o padrão é independente da escolha da velocidade do nodo, quando não é considerado tempo de pausa. Porém, quando considerado o tempo de pausa, o aumentando desta faz com que a distribuição espacial apoxime-se da distribuição uniforme.

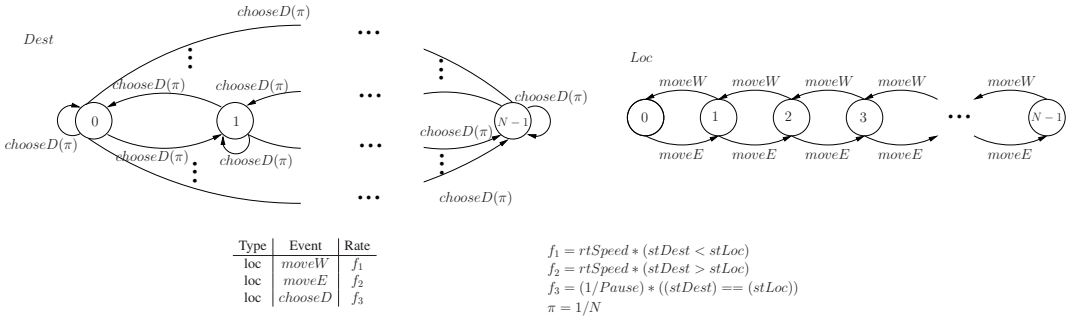
Na literatura, são encontrados vários estudos sobre o padrão RWP. Alguns destes estudos são voltados à análise da distribuição espacial de nodos, tanto aplicando ferramentas de simulação [3], como modelos analíticos [4, 7, 18]. Um estudo aprofundado sobre a aplicação do

modelo RWP em simuladores é efetuado em [23], apontando um comportamento instável durante um período inicial de simulação, que poderia gerar resultados equivocados em uma análise de desempenho de redes. Em [24] e [15], são apresentadas propostas para evitar a instabilidade do padrão durante a simulação.

Neste capítulo, serão apresentadas propostas de modelagem do padrão *Random Waypoint* para obter a distribuição espacial de nodos, utilizando Redes de Autômatos Estocásticos, bem como uma análise detalhada deste padrão. São consideradas áreas de movimentação em uma e duas dimensões, velocidade e tempo de pausa constantes. Os resultados obtidos são comparados com resultados encontrados na literatura.

4.1 Modelo SAN para Random Waypoint em 1D

Considerando área com uma dimensão, foi definido um modelo SAN para representar o padrão de movimentação em uma região de M metros modelada como uma reta ou uma linha, onde um nodo pode mover-se para oeste ou leste. Nesta modelagem, foi necessário discretizar a área de movimentação, representando-a como um espaço discreto de N slots.



Type	Event	Rate
loc	$moveW$	f_1
loc	$moveE$	f_2
loc	$chooseD$	f_3

$f_1 = rtSpeed * (stDest < stLoc)$
$f_2 = rtSpeed * (stDest > stLoc)$
$f_3 = (1/Pause) * ((stDest) == (stLoc))$
$\pi = 1/N$

Figura 25 – Modelo SAN para Random Waypoint em 1D.

O modelo SAN, proposto para uma dimensão, é apresentado na Figura 25, sendo composto de dois autômatos, chamados *Dest* e *Loc*. O autômato *Dest* representa a escolha aleatória de destino. O autômato *Loc* representa a localização do nodo na área de movimentação. Os dois autômatos têm *N* estados e cada estado representa uma segmento de tamanho M/N metros da área de movimentação.

Basicamente, duas situações podem ocorrer neste modelo SAN: o nodo movimenta-se até o destino escolhido ou o nodo está parado em uma posição, aguardando para escolher novo destino (pausa).

O autômato *Dest* tem apenas um evento chamado $chooseD$ (escolhe um destino), o qual pode ocorrer somente quando o nodo já alcançou o destino. Esta situação é representada com os dois autômatos no mesmo estado. O tempo de pausa, que ocorre quando o nodo alcançou o destino, é representado na taxa do evento $chooseD$, com o inverso do tempo de pausa

$(1/pausa)$. Além disso, a ocorrência deste evento é condicionada à igualdade dos estados dos dois autômatos. A taxa funcional do evento $chooseD$ está expressa na função f_3 (Equação 4.1) da Figura 25.

$$f_3 = \frac{1}{pausa} * (st\ Dest == st\ Loc) \tag{4.1}$$

Esta função retorna $1/pause$ quando o modelo está em situação de pausa, ou zero se o modelo está em situação de movimento. Além disso, para representar a escolha de todos os destinos possíveis existem transições entre todos os pares de estados do autômato $Dest$, ou seja, de qualquer estado em $Dest$ há N transições (evento $chooseD$) que levam a cada um dos estados de $Dest$. Para representar corretamente esta transição no modelo SAN, cada evento $chooseD$ tem a probabilidade π de ocorrer. Assumindo igual probabilidade de ocorrência para todas as direções:

$$\pi = \frac{1}{N} \tag{4.2}$$

O autômato Loc tem dois possíveis eventos representando o movimento do nodo para oeste ($moveW$) ou para leste ($moveE$). Os eventos $moveW$ e $moveE$ somente poderão ocorrer quando o modelo estiver em situação de movimento, ou seja, quando os estados de Loc e $Dest$ são diferentes. Estes dois eventos representam a movimentação do nodo, portanto, eles devem ser habilitados e desabilitados conforme o sentido ao qual o nodo deverá se movimentar, ou seja, se estado de Loc é menor que estado de $Dest$ (o destino está a leste do nodo), então $moveE$ é habilitado (setado com a taxa que representa a velocidade de movimentação, $rtSpeed$) e $moveW$ é setado com zero. Desta forma, o nodo somente poderá movimentar-se para leste, conforme Equações 4.3 e 4.4.

$$moveW = rtSpeed * ((stDest) < (stLoc)) \tag{4.3}$$

$$moveE = rtSpeed * ((stDest) > (stLoc)) \tag{4.4}$$

O $rtSpeed$ é a taxa que representa a velocidade de movimentação do nodo e é calculado conforme a Equação 4.5.

$$rtSpeed = \frac{Speed}{SizeSlot} \tag{4.5}$$

4.1.1 Analisando o Modelo 1D

Os resultados gerados foram validados, comparado-os aos resultados gerados por Bettstetter, Resta e Santi em [4]. Segundo os autores, a probabilidade da distribuição espacial, para um modelo genérico de uma dimensão sem pausa, é obtida através da Equação 4.6.

$$f_X(x) = -\frac{6}{a^3}x^2 + \frac{6}{a^2}x \tag{4.6}$$

onde a é o tamanho da região de movimentação e x é a posição nesta região. A observação mais importante para esta equação é que esta representa a distribuição em uma região como uma função contínua. Desta forma, para comparar o modelo SAN com os resultados desta equação, foi necessário assumir um número discreto de espaços, $slots$.

No modelo SAN, apresentado neste trabalho, é considerada a situação de pausa, possibilitando analisar o comportamento da distribuição espacial em relação à variação de pausa e de velocidade. Isto não é possível através da Equação 4.6, pois esta não considera a pausa, e a velocidade somente causa impacto na distribuição quando a pausa for maior que zero.

Estes resultados, aqui apresentados, também foram validados com a distribuição resultante da Equação 4.7, que é apresentada por Resta em [18].

$$\begin{gathered} f(x) = p_{stat} + (1 - p_{stat})p + (1 - p_{stat})(1 - p)6x(1 - x), \\ \text{if } x \in [0, 1], \text{ and } f(x) = 0 \text{ otherwise,} \\ \text{where } p = \frac{t_{pause}}{t_{pause} + \frac{1}{3v}} \end{gathered} \tag{4.7}$$

onde p_{stat} é a probabilidade do nodo permanecer parado (neste caso, assume-se zero), t_{pause} é o tempo de pausa, v é a velocidade do nodo e x é a posição na área [0,1]. Neste caso, como o tamanho da área é um, a área será dividida em N partes iguais, de acordo com o número de $slots$ do modelo.

A primeira validação efetuada foi através da comparação dos resultados obtidos pela Equação 4.6 usando $a = 20$ (no modelo, a área é de 1000 metros, representada por 20 slots de 50 metros cada um, um tempo de pausa muito baixo, $0,001s$, e velocidade de $20\ m/s$). Considerando o mesmo número de slots, alterando o tempo de pausa para $60\ s$ e velocidade $32\ m/s$, foram comparados os resultados obtidos com a Equação 4.7. Como o tamanho da área na equação é um, a velocidade (v) usada nesta equação deve ser proporcional, então a velocidade é dividida em metros/segundo por 1.000, que é o tamanho da área.

Numericamente os resultados obtidos demonstram a mesma probabilidade com um erro em torno de 10^{-4}. Foram analisados, também, alguns resultados com variações no tempo de pausa (de $0,1$ até $0,00001$ segundos) e outras variações maiores, que não apresentaram impacto significativo nos resultados. Porém, a variação do número de slots causa um considerável impacto na precisão. A Figura 26 demonstra que a diferença nos resultados gerados pela SAN e o modelo dos [4] e [18], considerando variações de 4 a 30 slots, tem uma queda exponencial que chega a 10^{-5}.

Assumindo os resultados obtidos para um tempo de pausa pouco significativo ($<= 0,1$) para comparar com a Equação 4.6 ([4]), foi analisado o efeito de um tempo de pausa significativo ($> 0,1$) para comparar com a Equação 4.7. ([18]). A Figura 27 apresenta a probabilidade, considerando área de $1.000\ m$ em $20\ slots$, velocidade $32\ m/s$ e tempo de pausa $0,001$, 4,

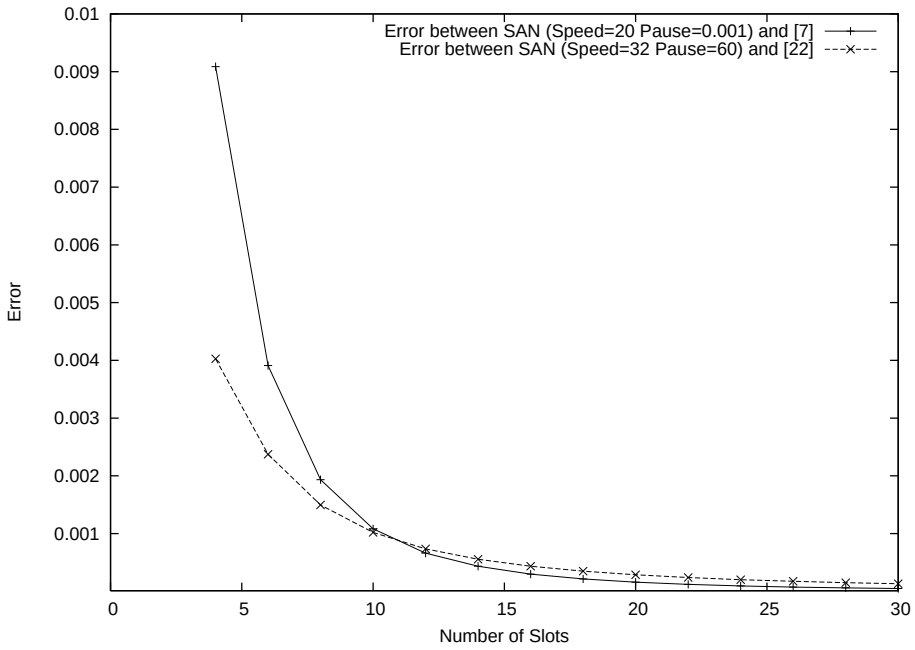


Figura 26 – Diferença entre modelos de Resta e SAN

64 e 256 segundos. Nesta figura, pode-se observar que o aumento no tempo de pausa faz os nodos se espalharem na área, ou seja, tende a distribuição uniforme. Este efeito demonstra que a importância, em relação à distribuição espacial, do tempo do nodo em cada $slot$ devido a sua passagem em movimento diminui, quando o tempo de pausa é aumentado. Assim, como o tempo do nodo em cada $slot$, decorrente do tempo de pausa, é uma distribuição uniforme, devido ao fato de que o nodo escolhe como destino todos os $slots$ com a mesma probabilidade, com o aumento no tempo de pausa, a distribuição tende a ser uniforme.

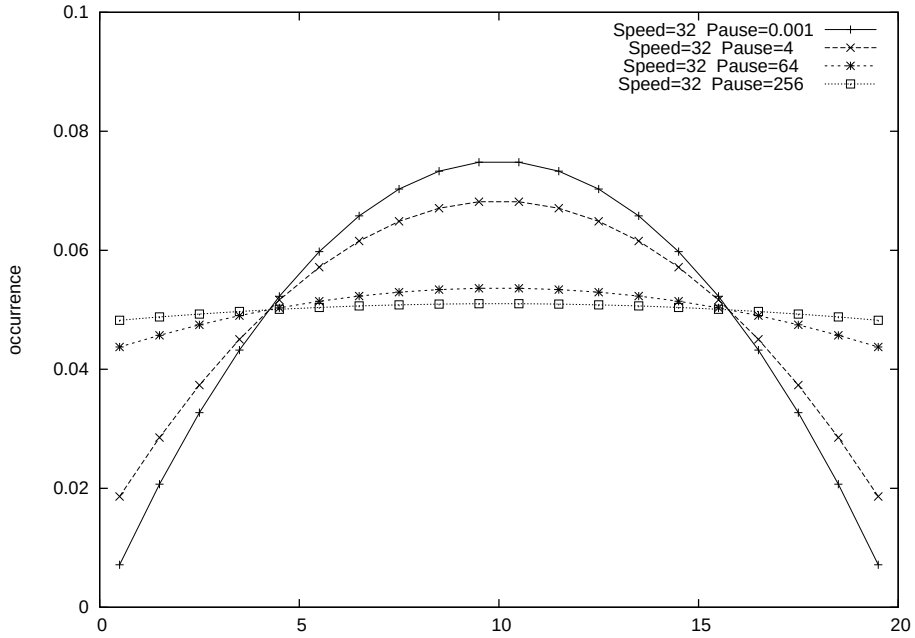


Figura 27 – Variação do tempo de pausa.

Outra característica analisada é a distribuição em relação à variação de velocidade. De acordo com [4] e [18], em conformidade ao descrito anteriormente, a variação na velocidade não causa efeito na distribuição quando a pausa é zero. Porém, quando considerada a pausa, a variação na velocidade causa efeito na distribuição. Na Figura 28 são apresentados resultados, considerando uma pausa significativa ($60\ s$) e variação de velocidade de 1, 4, 16 e $64\ m/s$. Estes resultados complementam as observações para o comportamento em relação às variações do tempo de pausa. Com uma maior velocidade, o tempo de passagem do nodo pelo $slot$ torna-se menos significativo na distribuição, tendendo a uma distribuição uniforme.

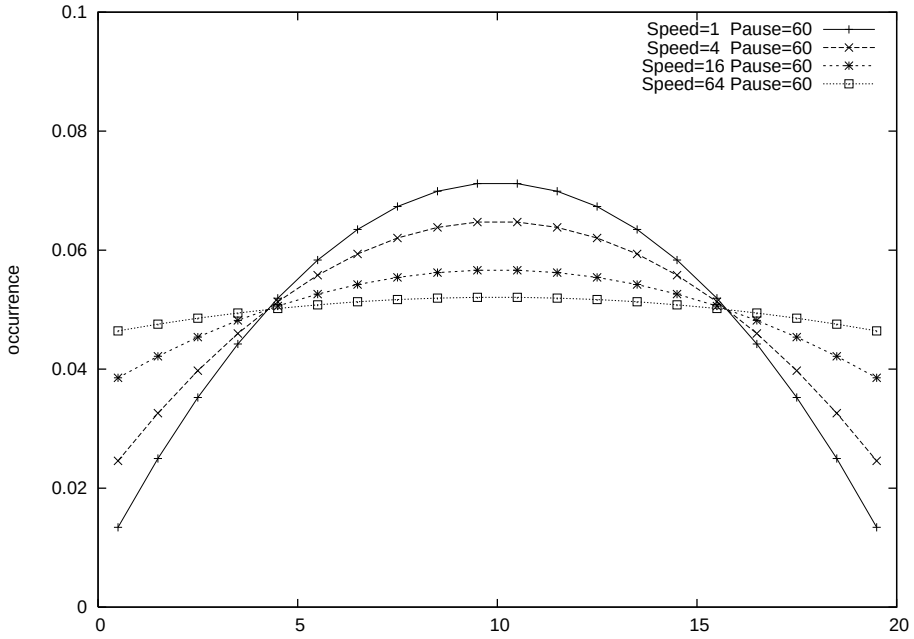


Figura 28 – Variação da velocidade do nodo.

4.2 Modelo SAN para Random Waypoint em 2D

O segundo modelo proposto é para representar a área de movimentação com duas dimensões, onde o nodo poderá movimentar-se em quatro direções[1]: oeste, leste, norte e sul. O modelo, que representa a movimentação nesta área, é apresentado na SAN da Figura 29 e é obtido por uma extensão natural da SAN para uma dimensão. A estrutura do modelo 1D é replicada, obtendo-se quatro autômatos que resultam no modelo 2D, onde há dois autômatos que representam a escolha de destino ($DestX$ e $DestY$) e dois autômatos que representam a movimentação do nodo ($LocX$ e $LocY$). Localização ou destino são representados por uma coordenada (x e y), representada pela combinação de autômatos $LocX$ e $LocY$ ou $DestX$ e $DestY$.

[1]Direção e sentido serão referenciados, neste trabalho, apenas como direção

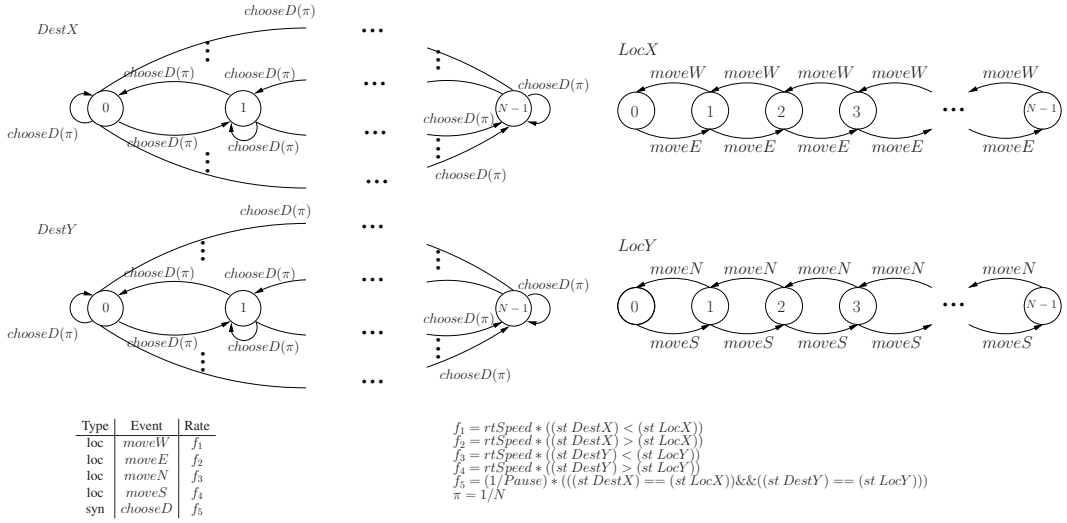


Figura 29 – Modelo SAN para Random Waypoint em 2D.

O comportamento das transições e eventos tem algumas diferenças em relação ao modelo 1D. A primeira destas é que a escolha de um novo destino é feita por eventos sincronizantes. Como descrito anteriormente, um destino é uma combinação de estados dos dois autômatos, $DestX$ e $DestY$, portanto a escolha do novo destino é um evento sincronizante que altera os estados destes dois autômatos simultaneamente, através do evento $chooseD$. Este evento é setado pela função f_5, conforme Equação 4.8, a qual representa o tempo de pausa e o momento em que pode ocorrer (quando o nodo chegou ao destino).

$$f5 = \frac{1}{Pause} * ((st\ DestX == st\ LocX)\ \&\&\ (st\ DestY == st\ LocY)) \tag{4.8}$$

Outra diferença entre os modelos 1D e 2D é a maneira como é representada a movimentação do nodo. No modelo 2D o movimento ocorre em quatro direções: oeste, leste, norte e sul (eventos $moveW$, $moveE$, $moveN$ e $moveS$, respectivamente). As taxas são setadas conforme a velocidade e direção que o nodo deve movimentar-se para alcançar o destino. Para definir a direção que o nodo deve seguir, é necessário definir uma estratégia de escolha de caminho.

4.2.1 Estratégias de Escolha de Caminhos

No modelo 1D, a escolha de direção, a fim de chegar ao destino, é muito simples, pois há somente duas possibilidades: os eventos $moveW$ ou $moveE$. No modelo 2D a escolha de direção para alcançar o destino, passa a ser mais complicada. O caminho direto entre a localização atual do nodo e o destino, é definida por um ângulo α, porém, neste modelo, devido a discretização da área de movimentação, os nodos podem andar em apenas quatro direções (moveW, moveE, moveN e moveS). Com isto, é necessário definir uma regra para escolha do caminho, de forma que o nodo alcance o destino, fazendo movimentos (através dos $slots$) nestas

quatro direções possíveis.

São apresentadas três estratégias de escolha de caminho: caminho aleatório, primeiro x depois y e algoritmo glutão, conforme exemplo na Figura 30. Intuitivamente, é esperada uma diferença na distribuição espacial de nodos, conforme a escolha da estratégia.

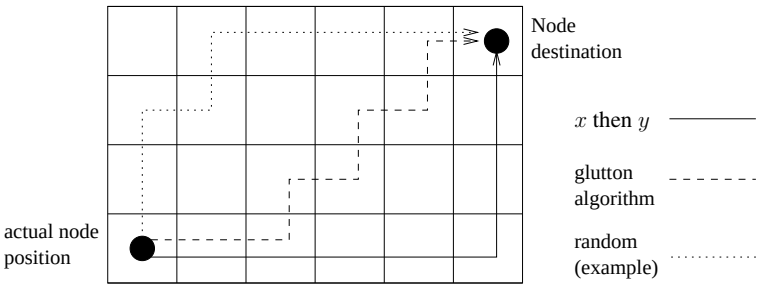


Figura 30 – Exemplo da estratégia de escolha de caminho.

A estratégia mais simples para ser representada em SAN é a decisão aleatória de caminho. Neste caso, a escolha do eixo para o movimento (x ou y, como $LocX$ ou $LocY$) é tomada estocasticamente entre os dois eventos possíveis (para alcançar um destino, o nodo terá no máximo dois eventos possíveis, os outros dois levariam à direção contrária). O modelo na Figura 29 representa esta opção. Transições de $LocX$ ($moveW$ e $moveE$) e $LocY$ ($moveN$ e $moveS$) somente estão habilitadas se devem realmente ocorrer, por exemplo, $moveW$ somente deve ocorrer se estado de $DestX$ é menor que estado de $LocX$ (destino está a oeste da localização atual do nodo).

A representação da estratégia de caminho, onde primeiramente é percorrido o eixo x depois o eixo y, é mais elaborada. Para isto é necessário alterar as taxas funcionais dos eventos de movimentação, incluindo a restrição de que $moveN$ e $moveS$ somente podem ocorrer, se os estados de $LocX$ e $DestX$ são iguais. As taxas funcionais, são setadas como:

$$\begin{aligned} f3 &= rtSpeed * (((stDestY) < (stLocY)) \;\&\&\; ((stDestX) == (stLocX))) \\ f4 &= rtSpeed * (((stDestY) > (stLocY)) \;\&\&\; ((stDestX) == (stLocX))) \end{aligned} \tag{4.9}$$

A representação da estratégia do *Algoritmo Glutão* é ainda mais elaborada e corresponde em alterar as taxas funcionais dos quatro eventos de movimentação. Neste caso, ocorre o evento do autômato que possui maior diferença entre a localização e destino. As funções dos eventos $moveW$, $moveE$, $moveN$ e $moveS$ são, respectivamente:

$$
\begin{aligned}
f1 &= rtSpeed * (((stDestX) < (stLocX)) \;\&\&\; (((stLocX) - (stDestX)) >= \\
&\quad (max((stDestY) - (stLocY), (stLocY) - (stDestY))))) \\
f2 &= rtSpeed * (((stDestX) > (stLocX)) \;\&\&\; (((stDestX) - (stLocX)) >= \\
&\quad (max((stDestY) - (stLocY), (stLocY) - (stDestY))))) \\
f3 &= rtSpeed * (((stDestY) < (stLocY)) \;\&\&\; (((stLocY) - (stDestY)) >= \\
&\quad (max((stDestX) - (stLocX), (stLocX) - (stDestX))))) \\
f4 &= rtSpeed * (((stDestY) > (stLocY)) \;\&\&\; (((stDestY) - (stLocY)) >= \\
&\quad (max((stDestX) - (stLocX), (stLocX) - (stDestX)))))
\end{aligned}
\tag{4.10}
$$

Nenhuma das estratégias de escolha de caminho, para um espaço discreto, deverá resultar no resultado preciso. Porém, é possível obter uma boa aproximação do comportamento em espaço contínuo utilizando a estratégia do algoritmo glutão, especialmente, quando considerado um grande número de slots.

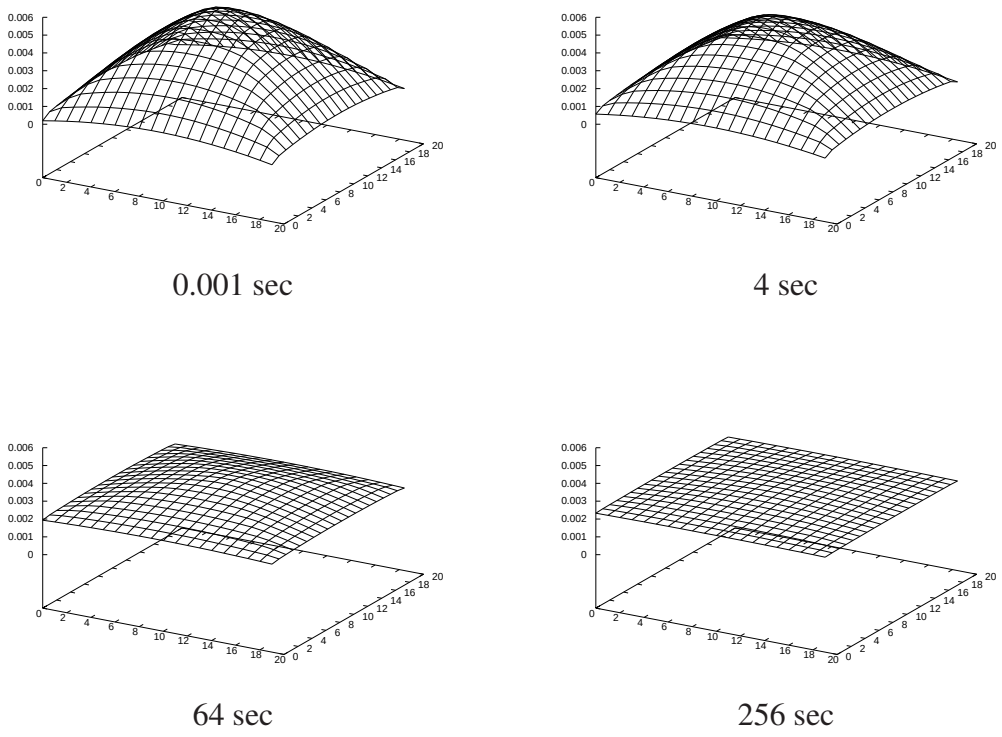


Figura 31 – Variação do tempo de pausa.

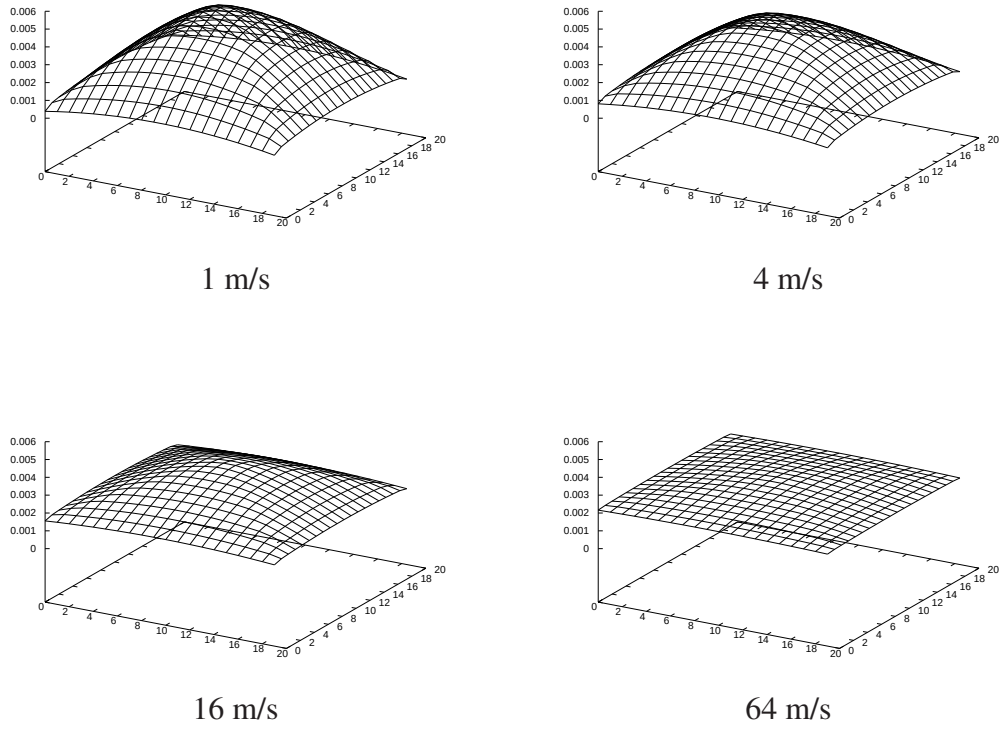


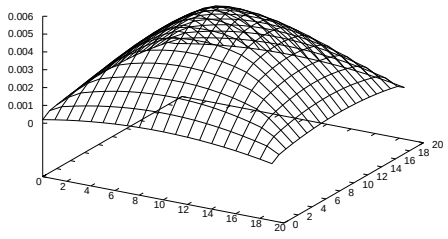
Figura 32 – Variação da velocidade.

4.2.2 Analisando o Modelo 2D

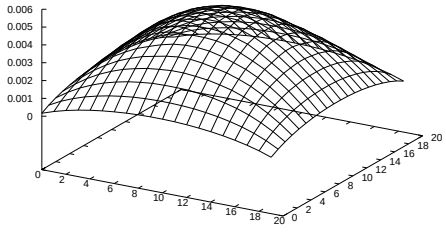
Analogamente ao modelo 1D, a validação do 2D foi efetuada comparando-se aos resultados de [4] e [18]. Para tanto, foi comparado modelo considerando pausa igual a zero (nesta proposta de modelagem sempre há pausa, então é considerada um tempo de pausa muito baixo: $0,001\ s$), área de $1.000\ m$, 20×20 slots e algoritmo glutão. A diferença entre resultados obtidos e apresentados por Bettstetter e Resta é em torno de 10^{-4}.

Outro comportamento observado é a variação na precisão em relação ao número de slots da área de movimentação, uma vez que está discretizada. Variando o número de $slots$ (5×5, 10×10) foi obtida uma precisão de 10^{-2} e 10^{-3}, respectivamente.

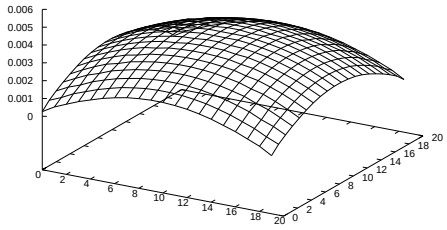
As variações de tempo de pausa e velocidade, feitas para o modelo 1D (Figuras 27 e 28), foram reproduzidas no modelo 2D usando o algoritmo glutão como estratégia de escolha de caminho. A Figura 31 demonstra a distribuição da probabilidade obtida para o modelo com velocidade constante ($32\ m/s$) e variando o tempo de pausa em $0,001$, 4, 64 e 256 segundos. Analogamente, na Figura 32 é demonstrado o resultado para a pausa fixada em 60 segundos e variando a velocidade em 1, 4, 16 e $64\ m/s$. Estes resultados demonstram claramente um comportamento equivalente ao do modelo 1D, ou seja, com tempo de pausa relevante, o aumento na velocidade, faz com que a distribuição espacial tende a ser uniforme.



Algoritmo Glutão



Caminho Aleatório



Primeiro o X depois o Y

Figura 33 – Efeito da escolha da estratégia para escolha de caminho.

Outra análise efetuada, foi a comparação do uso de diferentes estratégias de escolha de caminho. A Figura 33 demonstra a distribuição espacial encontrada para as três estratégias apresentadas anteriormente (aleatória, x depois y, algoritmo glutão) para o modelo com área de $1.000 \times 1.000\ m^2$, 20×20 slots, velocidade de $5\ m/s$ e tempo de pausa de $0,001$ segundos. Ao contrário dos outros experimentos, que não causavam deformação na curva da distribuição, apenas aumentavam ou diminuíam a concavidade, a variação da estratégia causa deformação na curva da distribuição espacial. Como era esperado, a estratégia, a qual foi encontrado maior efeito sobre a distribuição, é a que percorre primeiro o segmento do eixo X, depois o segmento do eixo Y, que tende a diminuir a concentração de nodos no centro da área.

5 Padrão Random Direction

O modelo *Random Direction* (*RD*) é um modelo baseado na escolha aleatória do sentido para o movimento, diferentemente, do modelo *Random Waypoint*, que é baseado, na escolha do destino a ser alcançado pelo movimento.

O modelo RD pode ser encontrado na literatura com algumas variações nas suas características. Entre os modelos encontrados, está o modelo apresentado por Royer, em [19]. Neste modelo, o nodo é, inicialmente, colocado em algum ponto da área, escolhe um sentido no intervalo de $[0..2\pi)$ e uma velocidade. Então inicia o movimento, que deve seguir até que o nodo alcance o limite da área de movimentação (borda). Alcançada a borda, o nodo espera por um tempo (pausa) antes de escolher novo sentido e velocidade para um novo movimento. Este novo sentido, após alcançada a borda, deve ser escolhido no intervalo de $[0..\pi]$, de forma que $\pi/2$ seja perpendicular à borda a qual o nodo se encontra e para dentro da área de movimentação. Desta forma, o nodo escolhe um sentido, que o mantém dentro da área de movimentação.

Em [9] e [16] os autores utilizam uma variação deste padrão, que também emprega a borda da área de movimentação como o limite do movimento. Neste padrão, a velocidade do movimento é constante, o sentido inicial é escolhido com igual probabilidade para todos os sentidos e ao alcançar a borda o nodo escolhe novo sentido. Assim como no modelo utilizado por Royer, ao incidir sobre a borda os dispositivos devem escolher um novo ângulo de forma a voltarem para dentro da área de movimentação. Porém, neste modelo a escolha deste ângulo (ângulo de reflexão) é feita em relação ao ângulo de incidência do dispositivo na borda. O ângulo de reflexão na borda é igual a $\pi - \beta$, sendo β o ângulo que o nodo atingiu a borda.

Royer ainda apresenta uma variação do seu padrão, *Modified Random Direction*, que é analisada também pelo Bettstetter em [3]. Neste padrão, sentido e velocidade variam a cada movimento. O sentido é escolhido no intervalo $[0..2\pi)$, sem qualquer preferência de valor, ou seja, com probabilidade igual para todos os valores. A velocidade pode ser escolhida conforme uma distribuição uniforme ou normal. Os movimentos são limitados por uma distância percorrida ou por um tempo de execução. Desta forma, o movimento não vai necessariamente até a borda. O tamanho do movimento é escolhido em uma distribuição exponencial. Pode-se ainda, considerar a existência de pausa, que é o tempo que o nodo espera para iniciar cada novo movimento.

De forma genérica, o padrão RD é descrito com a execução dos seguintes passos:

1. Escolhe sentido no intervalo de $[0..2\pi)$, essa escolha pode ser com igual probabilidade para os sentidos possíveis, por uma distribuição (normal, exponencial, ...) ou ainda definida pelo ângulo de incidência sobre a borda (no caso do padrão utilizado em [9]

e [16]) ;

2. Escolhe velocidade, conforme uma determinada distribuição;

3. Executa movimento. Este movimento pode ser limitado por tempo, distância ou pela borda;

4. Finalizado o movimento, o nodo pode ou não efetuar uma pausa. O tempo de pausa pode ser constante ou variar de acordo com alguma distribuição;

5. Inicia o processo novamente, retornando ao passo 1.

Como foi descrito, este padrão baseia-se na escolha de sentido para movimento, porém algumas características podem variar conforme necessidade de uso, ou para representar melhor a realidade estudada, como:

- Existência ou não da pausa;
- Velocidade constante ou escolhida em uma determinada distribuição;
- Limite de um movimento: o movimento pode ocorrer por um período de tempo, até que o nodo percorra uma distância ou até que o nodo alcance a borda;
- Regras de Bordas, que serão descritas na seção 5.1.

5.1 Regras de Bordas

No *Random Direction*, o dispositivo móvel executa cada movimento por um determinado tempo, distância ou até a borda, desta forma, o nodo poderá chegar ao limite da área de movimentação e não ter finalizado o movimento. Quando isto ocorre o nodo deve definir a seqüência do movimento de acordo com a regra de borda. Em [3], Bettstetter apresenta três diferentes regras de borda: *Bounce*, *Delete end Replace* e *Wrap Around*. Estas três regras de borda garantem que o número de nodos na simulação seja constante, o que é freqüentemente requerido nas simulações para avaliação de desempenho de redes.

1. *Bounce*: quando o nodo atinge a borda ele deve voltar para a área de movimentação, partindo do mesmo ponto onde se encontra, passará a executar um movimento M'. Esse movimento M' pode manter características do movimento anterior, mantendo a velocidade e escolhendo um ângulo de reflexão, como: $\pi - \beta$, sendo β o ângulo que o nodo atingiu a borda, ou ainda, M' pode ser um novo movimento, com escolha de velocidade e sentido, de forma a manter-se dentro da área de movimentação (fig. 34-1);

2. *Delete and Replace*: ao atingir a borda o nodo é colocado de volta na área de movimentação, podendo escolher qualquer ponto da área, com a mesma probabilidade (fig. 34-2);

3. *Wrap around*: ao atingir a borda, o nodo continua o movimento com mesmo sentido e velocidade, porém na borda oposta da área de movimentação (fig. 34-3).

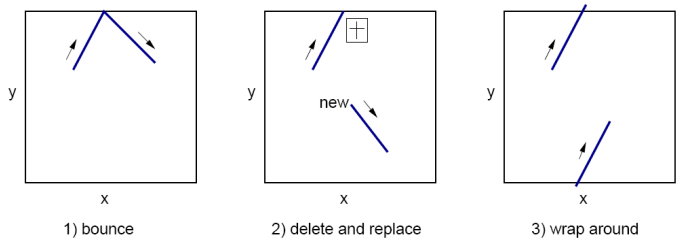


Figura 34 – Regras de bordas. O nodo alcança o limite da área de movimentação, deve tomar uma ação.

5.2 Modelo SAN Random Direction em 1D

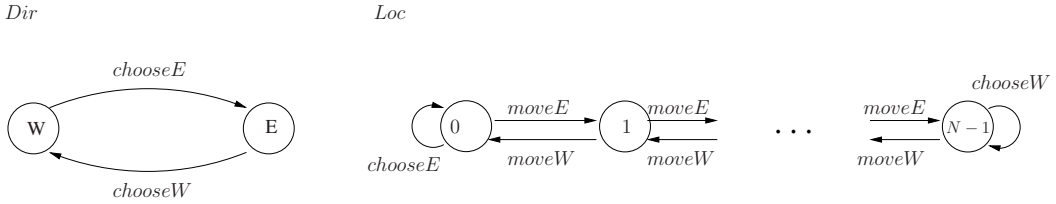
Nesta seção serão apresentadas propostas de modelagem, utilizando *SAN*, para análise da distribuição espacial de nodos para o padrão de mobilidade *Random Direction*. Nesta representação serão consideradas área de movimentação com uma dimensão e de livre movimentação (sem obstáculos), velocidade constante e pausa constante (também considerando que o tempo de pausa pode ser nulo). Nas seções a seguir serão apresentadas as propostas de modelagem para o modelos de *Royer* e para as três regras de bordas apresentadas na seção 5.1.

5.2.1 SAN para Padrão de Royer

O primeiro modelo a ser analisado é o padrão *Random Direction* utilizado por Royer em [19]. A principal característica deste modelo é que o movimente é executado até que o dispositivo alcance o limite da área de movimentação. Quando isto ocorre o nodo inicia outro movimento, de forma a permanecer dentro da área. Nesta primeira representação, não será considerado o tempo de pausa entre um movimento e outro.

Assim como no modelo apresentado para representar a padrão de mobilidade *Random Waypoint*, no Capítulo 4, no modelo, aqui definido, há um autômato para representar a região de movimentação, de M metros, modelada como uma linha, onde o nodo por mover-se para leste ou para oeste. Nesta modelagem, esta região de movimentação foi assumida como um espaço discreto de N $slots$ (numerados de 0 a $N-1$). Cada $slot$ representa uma região de $\frac{M}{N}$ metros.

O autômato *Loc* tem N estados, onde cada estado representa um segmento ($slot$) da área de movimentação. Um $slot$ representa uma N-ésima parte da área de movimentação. Cada estado



Type	Event	Rate
loc	$moveE$	f_1
loc	$moveW$	f_2
syn	$chooseE$	f_2
syn	$chooseW$	f_1

$rtSpeed = Speed/SizeSlot$
$f_1 = (stDir == E) * rtSpeed$
$f_2 = (stDir == W) * rtSpeed$

Figura 35 – Modelo SAN para Random Direction, modelo de Royer, sem considerar pausa.

possui dois evento de saída: ***moveW*** e ***moveE***, representando a movimentação do nodo para o próximo $slot$ a oeste ou a leste do atual. A taxa de ocorrência deste dois eventos é setada com $rtMove$, Equação 5.1, que representa a velocidade a qual o nodo percorre o segmento. Porém, ainda há um teste nesta taxa para saber se o nodo está se movimentando para leste ou para oeste, de forma a habilitar somente uma das duas direções, isto é feito através do do autômato Dir que será descrito logo a seguir.

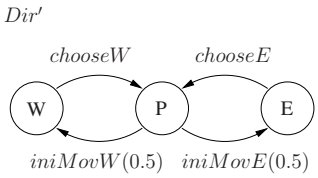
$$rtMove = \frac{Speed}{SizeSlot} \tag{5.1}$$

Os nodos das extremidade, que representam o último $slot$ a oeste ou a leste, da área de movimentação, possuem apenas o evento que traz o nodo de volta para área de movimentação, ou seja, para o estado a oeste (0), existe apenas o evento $moveE$ e para o estado a leste ($N-1$), o evento ***moveW***. Além disso, estes estados, 0 e $N-1$, possuem eventos que representam que o nodo alcançou a borda e deve voltar para dentro da área. Esses eventos são: ***chooseW***, para $N-1$, e ***chooseE***, para 0; são setados com os mesmos valores de $moveE$ e $moveW$, conforme Equações 5.2 e 5.3, e são sincronizantes.

$$chooseW = moveE \tag{5.2}$$

$$chooseE = moveW \tag{5.3}$$

Na modelagem apresentada no Capítulo 4 e em [7], para o RWP em 1D, foi utilizado um segundo autômato para representar a escolha do destino para o próximo movimento. No RD, também será necessário um segundo autômato para definir sobre o próximo movimento, porém, este autômato representa a escolha do sentido do movimento, para leste (E) ou oeste (W). O autômato ***Dir***, representa o sentido ao qual o nodo escolheu para efetuar um movimento, possuindo apenas dois estados: oeste (W) e leste (E). Desta forma, os eventos $moveE$ e $moveW$, estarão habilitados ou não, conforme o estado do autômato Dir. A mudança de estado do autômato Dir ocorre através dos eventos $chooseW$ ou $chooseE$, e somente poderá ocorrer quando o autômato Loc estiver no estado 0 ou $N-1$, ou seja, quando o nodo alcança a borda



Type	Event	Rate
loc	$iniMoveE$	f_3
loc	$iniMoveW$	f_4

$$rtIniMove = 1/Pause$$
$$f_3 = (stLoc == S_0) * rtIniMove$$
$$f_4 = (stLoc == S_{n-1}) * rtIniMove$$

Figura 36 – Autômato Dir', inclui a pausa no modelo de Royer.

da área, conforme modelo da Figura 35.

Como foi descrito anteriormente, o modelo da Figura 35 não representa a pausa, porém esta característica pode ser incluída substituindo o autômato Dir, pelo autômato Dir' apresentado na Figura 36. Neste autômato é incluído o estado P, que representa a pausa, ou seja, o autômato Loc não pode alterar seu estado, enquanto Dir' estiver no estado P. Assim, ao finalizar um movimento, antes de alterar o sentido, é preciso passar pelo estado P. E a alteração do estado P para um sentido, W ou E, ocorre através dos eventos $iniMoveE$ e $iniMoveW$, assim representando o tempo de pausa entre um movimento e outro.

A taxa de ocorrência dos eventos $iniMoveW$ e $iniMoveE$ representam o tempo de pausa que o nodo deve efetuar entre um movimento e outro, conforme a Equação 5.4.

$$rtIniMove = \frac{1}{Pause} \tag{5.4}$$

5.2.2 SAN para regra de borda Bounce

No modelo da seção anterior, o nodo, ao alcançar a borda, inicia novo movimento com nova velocidade e sentido. Segundo a regra de borda *bounce* (seção 5.1), o nodo, ao alcançar a borda, não inicia um novo movimento. Ele mantém a mesma velocidade, não efetua pausa e apenas define o novo sentido, de forma a manter-se dentro da área de movimentação. Este novo sentido é definido em relação ao ângulo ao qual o nodo alcançou a borda, ou seja, $\pi - \beta$, sendo β o ângulo de incidência do nodo na borda. Este novo sentido também pode ser escolhido por outra regra, como escolha aleatória.

Considerando área de movimentação com uma dimensão e regra de borda *Bounce*, a regra para definição do ângulo do movimento M', será indiferente, visto que ao alcançar a borda o nodo terá apenas um sentido a escolher.

O modelo SAN, apresentado na Figura 37, é semelhante ao modelo anterior, porém, a partir de agora será considerado que o movimento é executado por um tempo ou por um distância, ao contrário do modelo anterior que o movimento era executado até que o nodo alcançasse a borda.

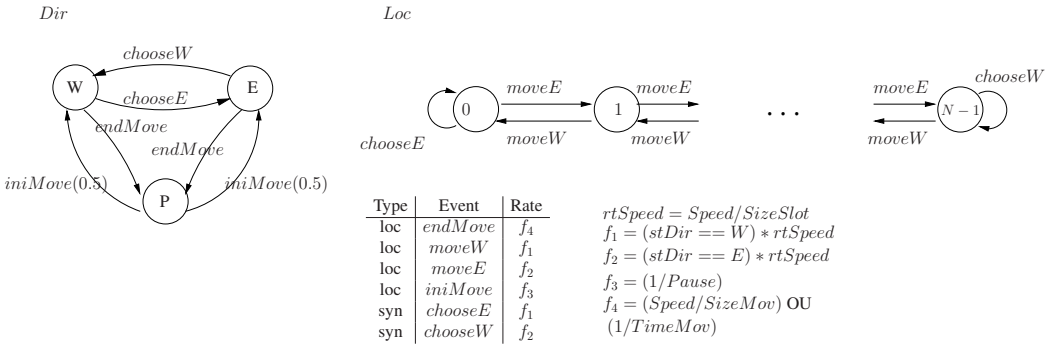


Figura 37 – Modelo SAN para Random Direction com regra de borda Bounce.

Nesta modelagem, os eventos $endMove$, que representam o fim de um movimento, podem ocorrer independente do $slot$ em que o nodo se encontra (autômato Loc). Assim, estes eventos ocorrem com uma taxa, de forma a representar o tamanho do movimento, setados conforme a Equação 5.5.

$$endMove = \frac{speed}{sizeMove} = \frac{1}{timeMove} \tag{5.5}$$

Os eventos $iniMove$ têm taxa igual ao inverso da pausa, modelando o tempo que o nodo aguarda antes de iniciar novo movimento. E, por fim, os eventos $chooseW$ e $chooseE$, são eventos sincronizantes, que ocorrem quando o nodo, ao alcançar a borda de da área de movimentação, deve mudar o sentido do movimento, conforme definido pela regra de borda *bounce*.

5.2.3 SAN para regra de borda Wrap Around

O terceiro modelo a ser representado (fig. 38), apresenta a regra de borda *Wrap Around*. Como descrito anteriormente, quando o nodo alcança a borda da área de movimentação, ele continua o seu movimento na borda da outra extremidade do área.

A regra de borda é modelada no modelo SAN utilizando eventos *moveE*, como evento de transição do estado S_{N-1} para S_0, e *moveW*, como evento de transição do estado S_0 para S_{N-1}. Lembrando que os eventos $moveW$ e $moveE$ são setados de forma a representar a velocidade de movimentação do nodo, conforme Equação 5.1.

Como pode ser visto na SAN, os eventos que modelam o efeito de borda, $moveW$ e $moveE$, ocorrem de forma independentes, ou seja, sem que ocorra qualquer alteração no movimento que o nodo está executando.

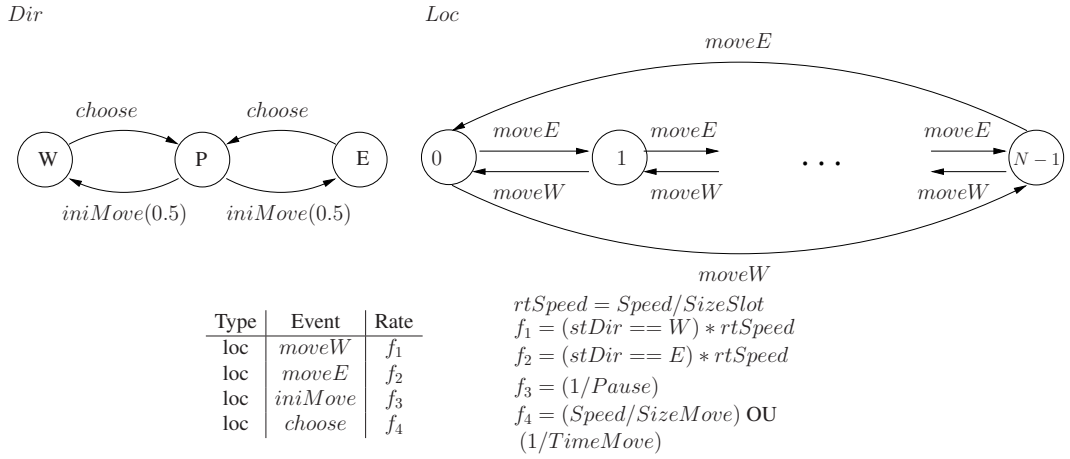


Figura 38 – Modelo SAN para Random Direction com regra de borda Wrap Around.

5.2.4 SAN para regra de borda Delete and Replace

O último modelo a ser apresentado, é para o padrão *Random Direction* com regra de borda *Delete and Replace*. Neste padrão, quando o nodo sai da área de movimentação em qualquer uma das bordas, ele deve ser retirado do experimento e imediatamente será inserido um novo nodo em algum ponto da área de movimentação.

Conforme é representado na Figura 39, para representar o efeito desta regra, foram criados dois novos eventos *delRepW* e *delRepE*. O evento *delRepW* pode ocorrer quando o autômato Loc está no estado 0 e Dir está em W, isto representa que o nodo está no $slot$ mais a oeste e está em movimento para oeste, assim alcançando a borda da área de movimentação. Desta forma, faz com que um dos eventos $delRepW$ ocorra, com taxa igual a $moveE/N$. Para cada estado de Loc existe um evento $delRepW$, originado no estado 0. O mesmo ocorre com o estado $N-1$ e os eventos $delRepE$, porém considerando que o movimento ocorre para leste.

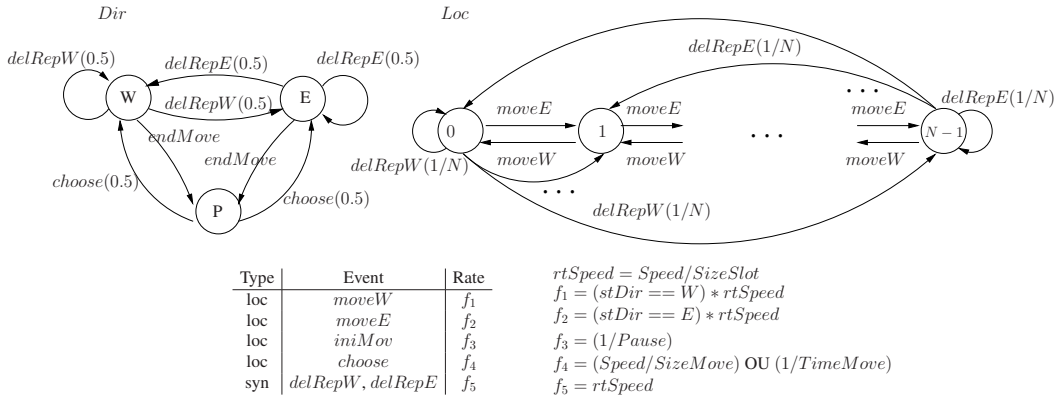


Figura 39 – Modelo SAN para Random Direction com regra de borda Delete and Replace.

5.2.5 Resultados para modelo em 1D

Nesta seção serão apresentados os resultados obtidos para a distribuição espacial de nodos em 1D, utilizando os modelos SAN apresentados nas seções 5.2.1, 5.2.2, 5.2.3 e 5.2.4, modelos de *Royer*, *Bounce*, *Wrap Around* e *Delete and Replace*, respectivamente.

Para validar os modelos propostos foi considerado uma região de 1.000 metros, dividida em 20 slots ($N = 20$), cada slot representa 50 metros. Na avaliação foram cosideradas variações de tempo de pausa, velocidade e tamanho do movimento.

O primeiro resultado a ser apresentado é para o padrão de movimentação de Royer, descrito na seção 5.2.1, utilizando o modelo sem considerar a possibilidade de pausa entre um movimento e outro. A distribuição espacial de nodos pode ser visto na Figura 40 e demonstra uma distribuição uniforme na área de movimentação, com área de uma dimensão, independente da velocidade escolhida para o movimento.

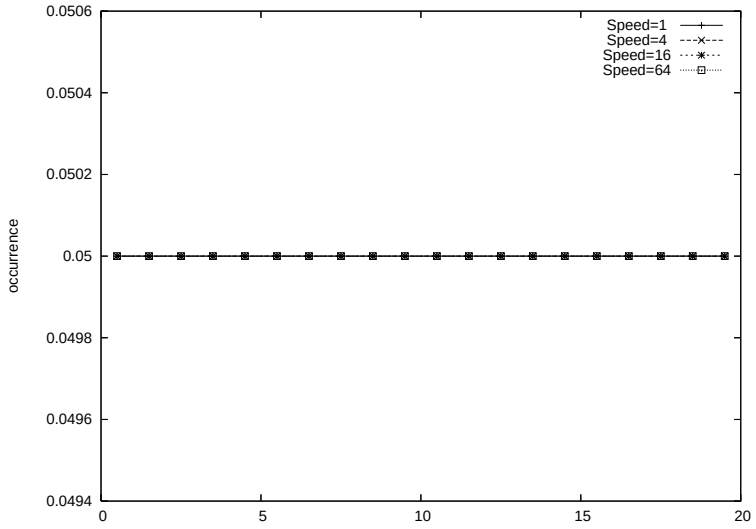


Figura 40 – Distribuição espacial de nodos para padrão de Royer sem pausa.

Ao inserir a pausa neste padrão, ele apresentará concentração de nodos nas extremidades da área de movimentação, como pode ser visto nos gráficos das Figuras 41 e 42. Este efeito deve-se ao fato de que a pausa ocorre entre o final de um movimento e início do próximo movimento, e, neste padrão, os movimentos sempre acabam na borda da área. Fazendo com que os nodos, executem a pausa na borda da área.

A concentração de dispositivos móveis em alguma região da área de movimentação, normalmente não ocorre em outros padrões de movimentação e pode ser indesejada, a não ser que seja a representação de alguma situação real ou outra característica desejada pelo autor. Mas, a discussão sobre os efeitos ou grau de realismo deste padrão não é foco deste trabalho, sendo representado este padrão, em *SAN*, considerando as duas possibilidades: sem pausa (fig.35) e com pausa (fig.36). A distribuição espacial de nodos, para o modelo com pausa, é apresentada

nos gráficos da Figura 41, considerando o efeito da variação de tempo de pausa, e na Figura 42, considerando o efeito da variação de velocidade.

Esta concentração de dispositivos ocorre exatamente na região da borda. Na análise, aqui apresentada, a concentração dos dispositivos ocorre nos $slots$ das extremidades, que representam os primeiros e os últimos $50m$ da área de movimentação. Isto ocorre, pois as bordas da área de movimentação foram modeladas como partes integrantes destes $slots$. Desta forma, independentemente do número de $slots$ utilizados na área de movimentação, a concentração de nodos ocorrerá no primeiro e no último $slot$ da área.

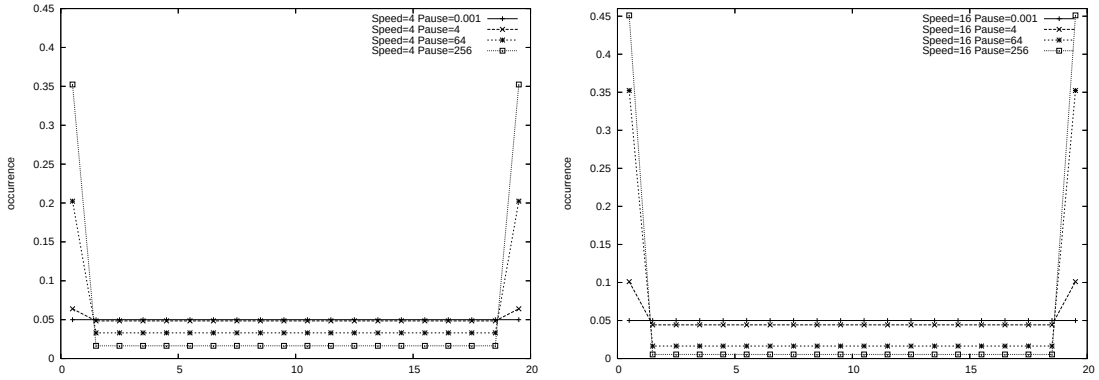


Figura 41 – Distribuição espacial de nodos para padrão de Royer com a pausa. Velocidade de $4m/s$ e $16m/s$ com variação da pausa.

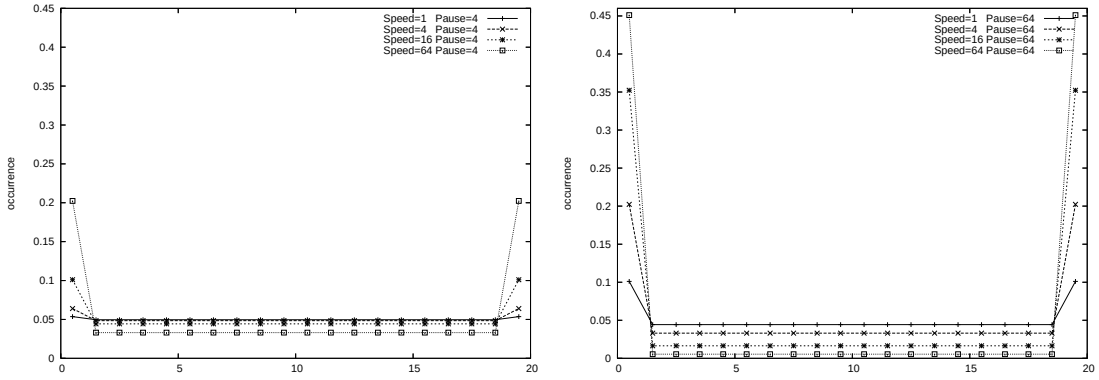


Figura 42 – Distribuição espacial de nodos para padrão de Royer com a pausa. Pausa de $4s$ e $64s$ com variação da velocidade.

Em [3], o autor analisa, através de simulação, a distribuição espacial de nodos, utilizando padrão de movimentação *Random Waypoint* e *Random Direction*, considerando área de movimentação em duas dimensões. Para *Random Direction*, são consideradas as regras de borda *bounce*, *wrap around* e *delete and replace*. Com os modelos, apresentados aqui, para representar o padrão *Random Direction*, com as variações de regra de borda: *bounce*, seção 5.2.2, e *wrap around*, seção 5.2.3, foi obtida uma distribuição uniforme, assim como a apresentada por Bettstetter na sua análise em duas dimensões. Os resultados obtidos podem ser vistos nas Figuras 43 e 44.

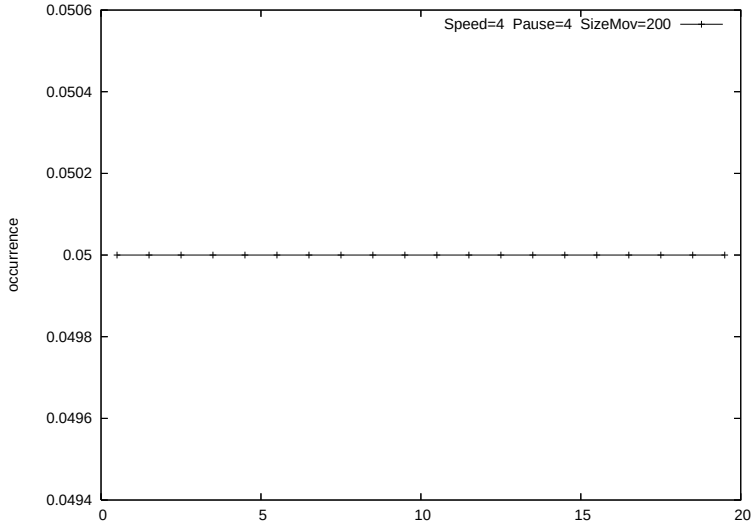


Figura 43 – Random Direction com regra de borda *bounce*.

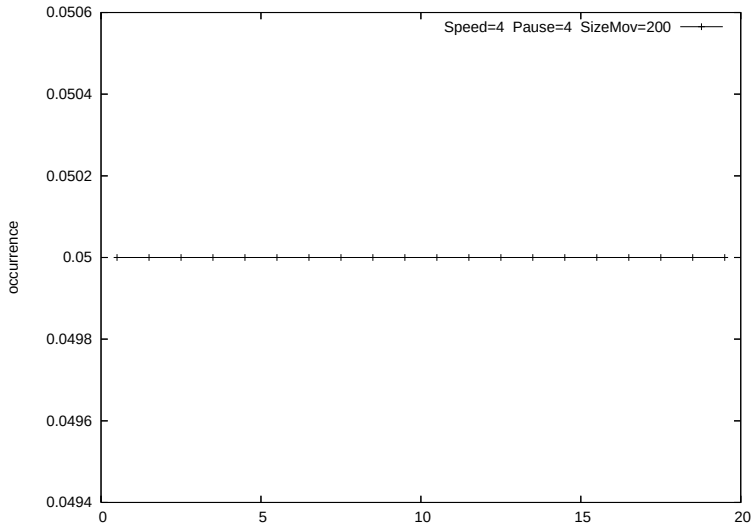


Figura 44 – Random Direction com regra de borda *Wrap Around*

Nos gráficos da Figuras 45 e 46, são apresentados os resultados obtidos considerando a regra de borda *delete and replace*, seção 5.2.4. Neste padrão, há uma concentração de nodos no centro da área de movimentação, a variação de velocidade e pausa não causam impacto na distribuição espacial de nodos, porém o tamanho do movimento influencia a distribuição, como será demonstrado a seguir.

No *RD* os movimentos são executados por um período de tempo ou por uma distância, quando utilizada a regra de borda *delete and replace*, a variação no tamanho do movimento (seja por tempo ou distância), causa efeito sobre a distribuição espacial dos nodos, como pode ser visto nas Figuras 47 e 48. Em [3], Bettstetter demonstrou que ocorre a curvatura no gráfico da distribuição espacial, quando utilizada esta regra de borda, bem como os efeitos de variação do tamanho de movimento.

Desta forma, é possível concluir que a pausa e velocidade não influenciam na distribui-

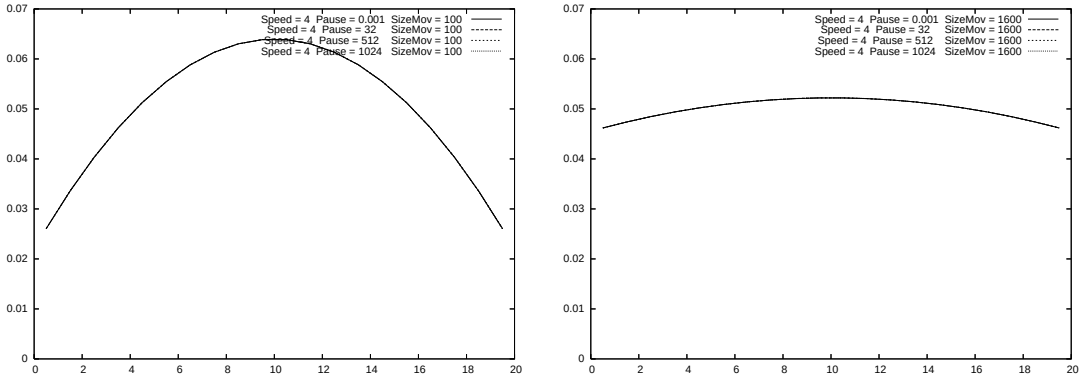


Figura 45 – *Random Direction* com regra de borda *Delete and Replace*. Variação de Velocidade e Pausa.

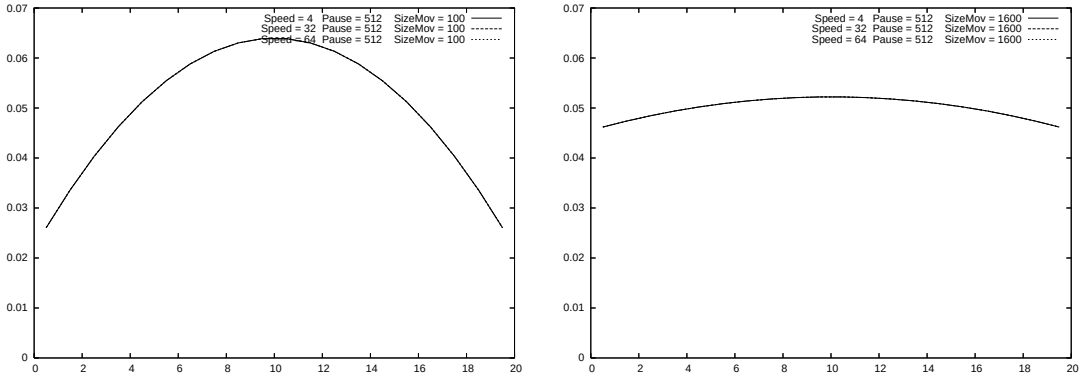


Figura 46 – *Random Direction* com regra de borda *Delete and Replace*. Variação de Velocidade e Pausa.

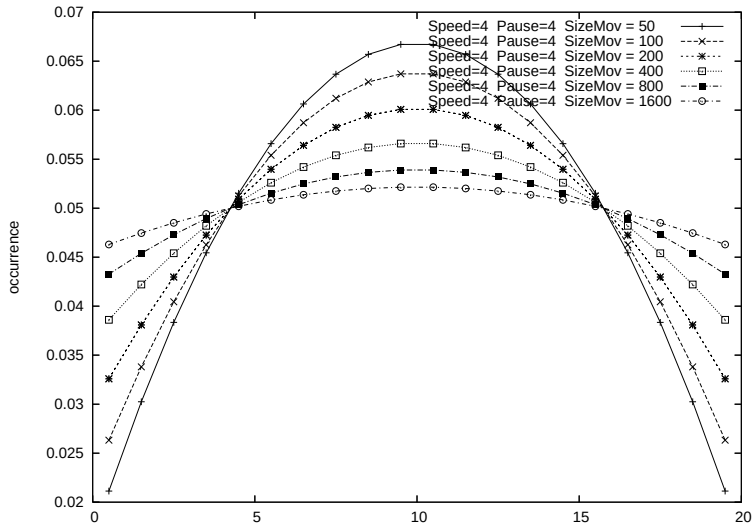


Figura 47 – *Random Direction* com regra de borda *Delete and Replace*. Velocidade de $4m/s$, pausa de $4s$ e variação do tamanho do movimento.

ção espacial de nodos, dependendo unicamente do tamanho do movimento: com aumento do tamanho do movimento a distribuição tende a ser uniforme.

Para todas as combinações de velocidade e tempo de pausa, fixando o tamanho do movimento, são obtidos os mesmo resultados. Aprofundando a observação, a distribuição espacial é

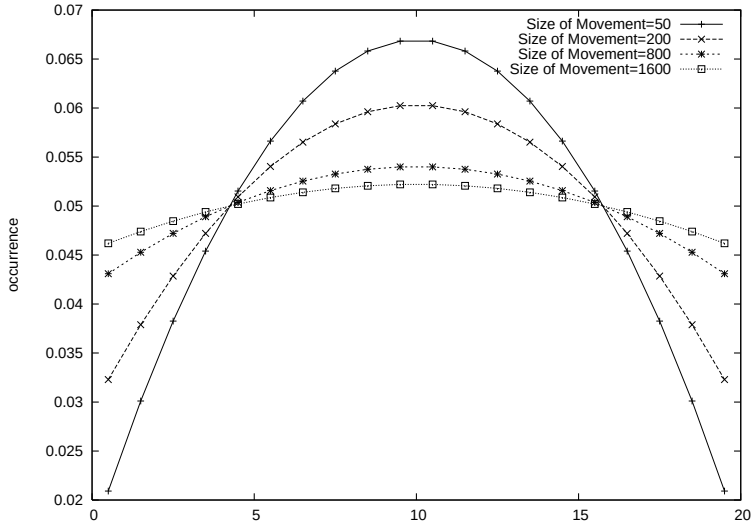


Figura 48 – Variação do tamanho do movimento para todas combinações de velocidade (1, 4, 16, 64 metros por segundos) e tempo de pausa (“0,001”, 16, 64 e 256 segundos).

uma combinação de tempos: tempo que o nodo efetua a pausa em cada *slot* e tempo que o nodo está em cada *slot* durante o seu movimento (passagem por uma região). A Figura 49 demonstra graficamente o comportamento da distribuição de localização do nodo em relação ao tempo de pausa e tempo em movimento, considerando as combinações de pausa $4s$ e velocidades de $4\ m/s$ e $32\ m/s$.

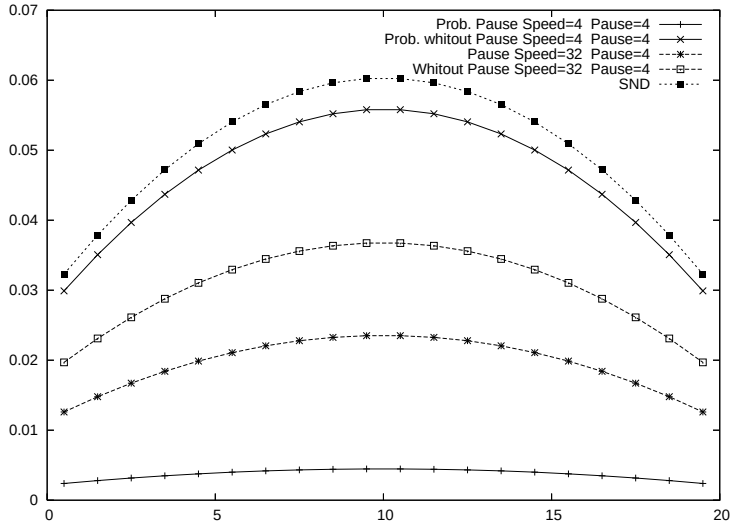
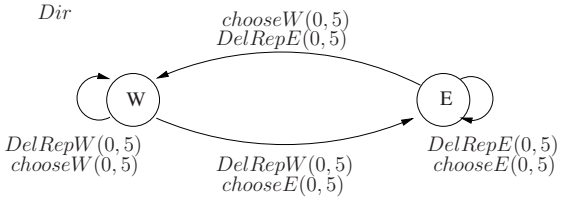


Figura 49 – Composição da distribuição espacial a partir da pausa e movimento.

Usando estes resultado, podemos simplificar o modelo SAN, o autômato Dir foi refeito de forma a não mais representar a pausa entre um movimento e outro. Ou seja, sempre que o nodo acabar de executar um movimento, ele iniciará um novo, no mesmo instante. Esta simplificação é feita mantendo o autômato Loc da Figura 39 e substituindo o autômato Dir por Dir' da Figura 50.

Esta nova versão de autômato Dir tem quatro eventos chamados $chooseW$, $chooseE$,



Type	Event	Rate	
loc	$chooseW, chooseE$	f_4	$f_4 = (Speed/SizeMove)OR(1/TimeMove)$
syn	$DelRepW$, $DelRepE$	f_5	$f_5 = rtSpeed/NumSlots$

Figura 50 – Autômato Dir', autômato Dir simplificado, sem o estado P, desconsiderando pausa entre movimentos.

$delRepW$ e $delRepE$. Os eventos $delRepW$ e $delRepE$ tem a mesma função que anteriormente, eles são usados para modelar o efeito de borda. Os eventos $chooseW$ e $chooseE$ tem também a mesma semântica como anteriormente. Porém, neste caso, não é considerada a ocorrência de pausa, nova direção é escolhida ao final de um movimento. As taxas dos eventos $chooseW$ and $chooseE$ são as mesmas que os eventos $endMove$, considerados anteriormente. A solução numérica para o modelo sem pausa, como esperado, é semelhante aos resultados obtidos com o modelo com pausa.

A Tabela 1 demonstra as diferenças numéricas entre os modelos sem pausa e com pausa, considerando pausa muito baixa de 0,1 seg. e 0,01 seg. A primeira coluna é a região (posição) da área de movimentação. A coluna SP contém os valores obtidos com o modelo sem pausa, a coluna $P1$ e $P2$ são valores obtidos no modelo com pausa. As colunas $SP - P1$ e $SP - P2$ demonstram a diferença do modelo sem pausa e modelo que consideram pausa (com valores relativamente baixos). Na última linha da tabela, estão os valores para as maiores diferença de cada uma das comparações (“Maior diff”). Pode-se observar que os valores são menores que 10^{-7}, considerando área de movimentação com 20 slots.

5.3 Modelo SAN para Random Direction em 2D

Na seção anterior foram apresentados os resultados obtidos para a distribuição especial de nodos para o *Random Direction*, considerando área de movimentação com uma dimensão. Agora, será apresentada uma proposta de modelagem SAN para este padrão, considerando área de movimentação com duas dimensões, onde os nodos podem se movimentar em várias direções. Considerando que, conforme observado na seção anterior, no modelo em 1D a variação de velocidade e pausa não impactam na distribuição espacial dos nodos, o modelo em 2D será construído sem considerar pausa entre movimentos, simplificando a modelagem e diminuindo o espaço de estados do modelo.

Considerando que, para área de movimentação em 1D e os resultados obtidos por Bettstetter em [3], os padrões com regra de borda *Wrap around*, *Royer* e *Bounce* apresentam distribuição

Pos	SP	P1 = 0,1s	SP - P1	P2 = 0,01s	SP - P2
1	0,046194927	0,046194929	-0,000000002094	0,046194921	0,000000005519
2	0,047396529	0,04739653	-0,000000000740	0,04739651	0,000000018922
3	0,04846462	0,048464619	0,000000000272	0,048464595	0,000000024947
4	0,049399199	0,049399198	0,000000000874	0,049399175	0,000000023750
5	0,050200267	0,050200266	0,000000001059	0,050200251	0,000000016488
6	0,050867824	0,050867823	0,000000000905	0,050867819	0,000000005092
7	0,051401869	0,051401869	0,000000000526	0,051401877	-0,000000008091
8	0,051802403	0,051802403	0,000000000072	0,051802424	-0,000000020627
9	0,052069426	0,052069426	-0,000000000324	0,052069456	-0,000000030353
10	0,052202937	0,052202937	-0,000000000551	0,052202973	-0,000000035647
11	0,052202937	0,052202937	-0,000000000551	0,052202973	-0,000000035647
12	0,052069426	0,052069426	-0,000000000324	0,052069456	-0,000000030353
13	0,051802403	0,051802403	0,000000000072	0,051802424	-0,000000020627
14	0,051401869	0,051401869	0,000000000526	0,051401877	-0,000000008091
15	0,050867824	0,050867823	0,000000000905	0,050867819	0,000000005092
16	0,050200267	0,050200266	0,000000001059	0,050200251	0,000000016488
17	0,049399199	0,049399198	0,000000000874	0,049399175	0,000000023750
18	0,04846462	0,048464619	0,000000000272	0,048464595	0,000000024947
19	0,047396529	0,04739653	-0,000000000740	0,04739651	0,000000018922
20	0,046194927	0,046194929	-0,000000002094	0,046194921	0,000000005519
		Maior diff	**0,000000002094**	**Maior diff**	**0,000000035647**

Tabela 1 – Comparação entre modelo sem pausa e modelo com pausa (0,1 s e 0,01 s).

uniforme, neste trabalho será tratado, para o padrão *Random Direction*, apenas a regra de borda *Delete and Replace* para o modelagem em 2D.

O padrão de movimentação em área com duas dimensõese será representado por três autômatos: *LocX*, *LocY* e *Degree*. Nesta modelagem foram reutilizadas algumas partes do modelo em 1D. No modelo 1D a representação da área de movimentação era feito pelo autômato *Loc*, aqui serão utilizados dois autômatos semelhantes ao *Loc*, chamados *LocX* e *LocY*, que representam os dois eixos de localização dos nodos. Desta forma, *LocX* representa a localização do nodo no eixo $oeste/leste$, e *LocY* no eixo $norte/sul$ (fig. 51). A superfície é dividida em $N \times N$ *slots* e a posição corrente do nodo é a coordenada dada pela combinação de estados dos autômatos $LocX$ e $LocY$.

Relembrando, o comportamento de borda *Delete and Replace* define que quando um nodo sai por alguma das bordas, imediatamente ele é reposicionado na área de movimentação. Neste caso, será considerado que a escolha da nova posição do nodo é equiprovável para todas as regiões (*slots*) da área.

Para modelar o comportamento de borda serão usados eventos chamados de $delRep$. Na Figura 52, foram incluídos os eventos $delRepW$, que modelam o comportamento que o nodo deverá ter ao alcançar a borda oeste da área, ou seja, quando o nodo alcançar a borda oeste

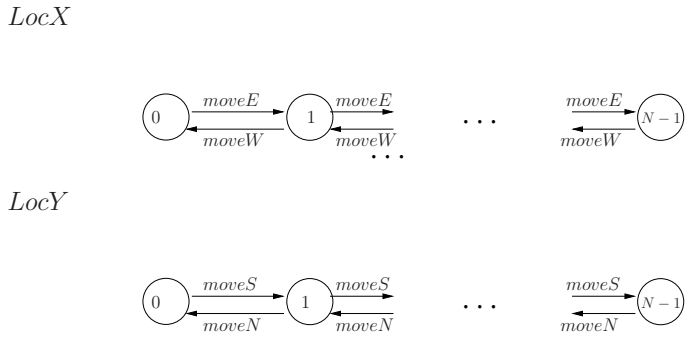


Figura 51 – Autômatos $LocX$ e $LocY$, representam a área de movimentação em 2D.

($LocX$ está no estado 0 e continua se movimentando para oeste) o estado de $LocX$ deverá ser alterado para um novo estado, que pode ser qualquer estado, inclusive o próprio 0. Como a localização do nodo é uma composição dos dois eixos, o estado de $LocY$ também deve ser alterado, para que o nodo reinicie o movimento em qualquer posição da área. Desta forma, o evento $delRepW$ é um evento sincronizante, que altera os estados de $LocX$ e $LocY$, simultaneamente.

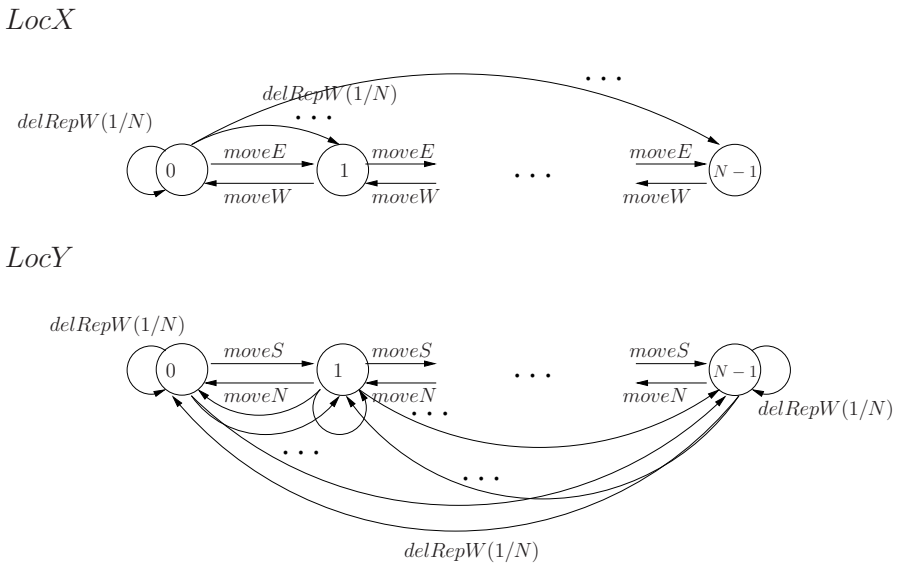


Figura 52 – Autômato LocX e LocY com efeito de borda a oeste. Modelo parcial para Random Direction com *Delete and Replace* em área 2D.

O comportamento da borda a leste da área de movimentação é modelado da mesma forma, utilizando o evento $delRepE$, como pode ser visto na Figura 53.

O comportamento das bordas norte e sul, são modelados com os eventos $delRepN$ e $delRepS$, respectivamente, demonstrado na Figura 54, onde é apresentado autômato completo para representar a localização do nodo na área de movimentação em 2D, inclusive representando o comportamento de borda. Para representar a movimentação do nodo, ainda, são precisos dois componentes: a velocidade e a direção do movimento.

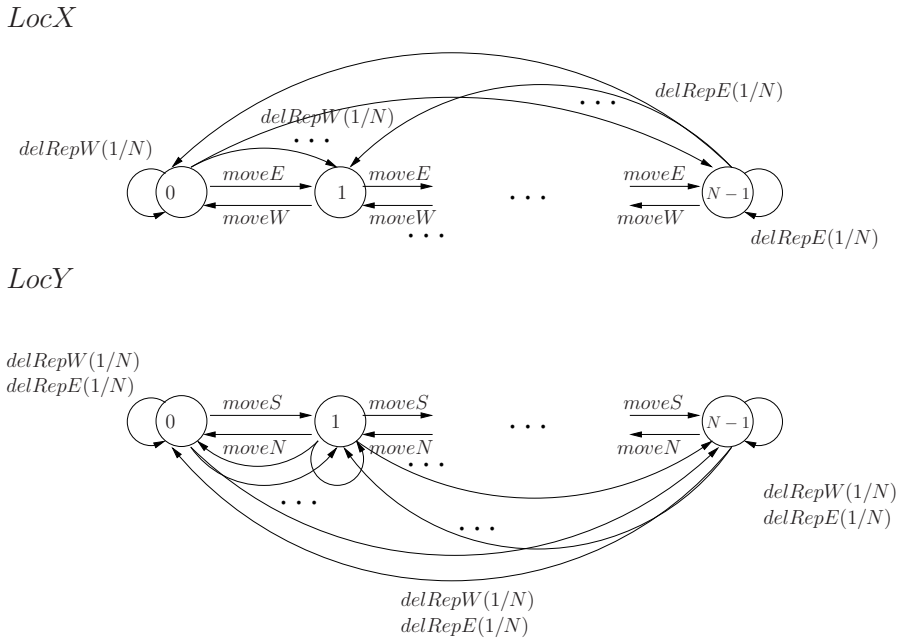


Figura 53 – Autômatos LocX e LocY com efeito de borda a oeste e a leste. Modelo parcial para Random Direction com *Delete and Replace* em área 2D.

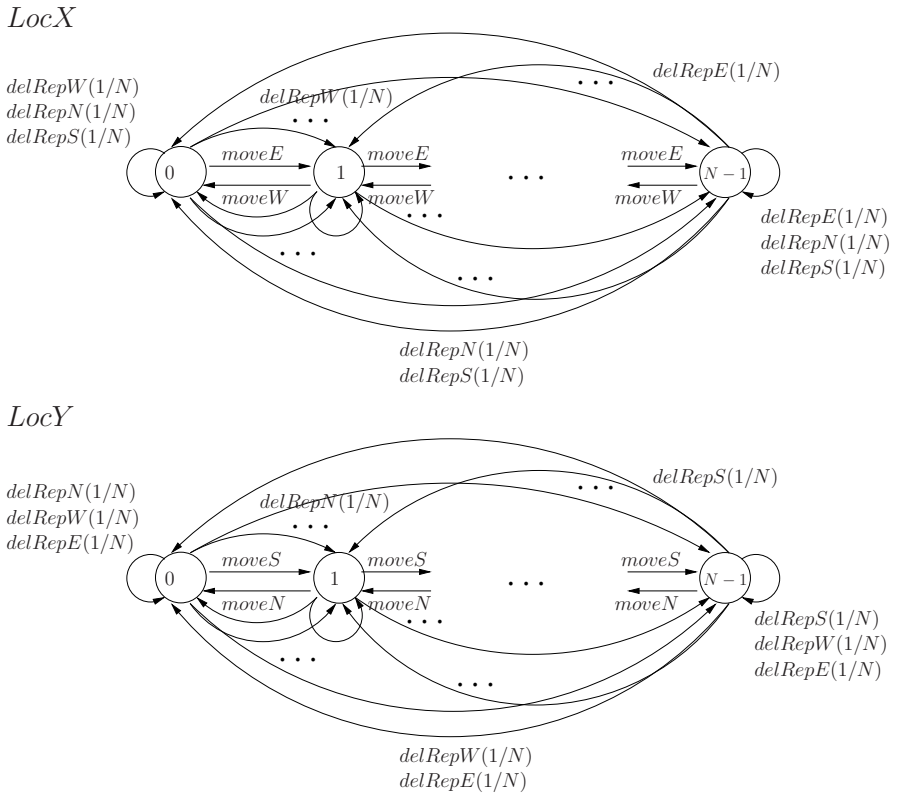


Figura 54 – Autômatos LocX e LocY completos para o padrão Random Direction com *Delete and Replace* em área 2D.

5.3.1 Velocidade e Direção para Random Direction em 2D

No padrão de movimentação *Random Direction* a direção do movimento é escolhida em um ângulo no intervalo $[0°, 360°)$, com igual probabilidade para todas as direções.

A escolha de direção foi modelada utilizado o autômato $Degree$ (fig. 55), que representa a escolha de um ângulo no intervalo de $[0°, 360°)$, de forma discretizada. Este autômato terá D estados, sendo que o primeiro estado é igual ao ângulo de $0 \times \frac{360°}{D}$, o segundo é igual a $1 \times \frac{360°}{D}$, o terceiro igual a $2 \times \frac{360°}{D}$ e assim por diante.

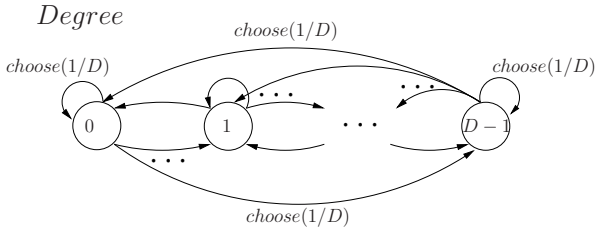


Figura 55 – Autômato $Degree$, modela a escolha de direção do movimento.

Nesta padrão de movimentação, o nodo efetua o movimento por um período de tempo ou por uma distância. Para modelar este comportamento, de final de movimento, foram usados os eventos $choose$. Em cada estado do autômato $Degree$ existe D eventos $choose$, que fazem a transição do estado atual para qualquer outro estado de $Degree$. Desta forma, o evento choose é setado com uma taxa que representa o tempo que o nodo deve executar o movimento, ou seja, o estado de $Degree$ é mantido pelo período necessário para o nodo executar o movimento.

Após escolher a direção para o movimento, é necessário fazer com que o nodo movimente-se conforme esta direção. Ou seja, que ocorra as transições de estados nos autômatos $LocX$ e $LocY$, de forma a representar o movimento a ser executado.

Os autômatos de localização consideram movimentos sobre os eixos, através dos eventos $moveW$ ou $moveE$ sobre o eixo $oeste/leste$ e $moveN$ ou $moveS$ sobre o eixo $norte/sul$. Desta forma, representar a movimentação do nodo sobre os eixos, ângulos de $0°$, $90°$, $180°$ ou $270°$, é direta, bastando habilitar um dos eventos $move$. Porém, a movimentação que necessitar deslocamento sobre os dois eixos requer maiores cuidados.

Para resolver este problema, é considerado que o movimento a ser executado é a hipotenusa de um triângulo retângulo, o qual os catetos adjacente e oposto são paralelos aos eixos $oeste/leste$ ou $norte/sul$. Desta forma, se os autômatos $LocX$ e $LocY$ efetuarem as transições de estados de forma a representar os tamanhos dos catetos, o nodo terá efetuado o movimento conforme e hipotenusa deste triângulo, que é o movimento, como pode ser visto na Figura 56.

Para possibilitar esse movimento, são calculados os catetos adjacente e oposto, de acordo com as Equações 5.6 e 5.7.

$$\cos \alpha = \frac{cateto\_adjacente}{hipotenusa} \tag{5.6}$$

$$\sin \alpha = \frac{cateto\_oposto}{hipotenusa} \tag{5.7}$$

Obtidos os tamanhos dos catetos, é necessário identificar qual dos eixos é cateto oposto e qual é o adjacente. Isto é feito de acordo com o quadrante que o movimento será executado,

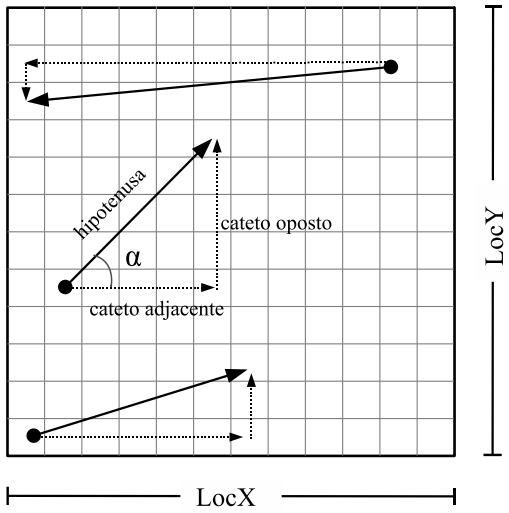


Figura 56 – O movimento a ser executado é a hipotenusa de um triângulo retângulo, formado pelos movimentos dos eixos oeste/leste e norte/sul.

conforme descrito na Figura 57. Também, é necessária a definição de qual evento $move$ deverá ocorrer, pois não podem ocorrer simultaneamente eventos com sentidos opostos ($moveW$ e $moveE$ ou $moveN$ e $moveS$).

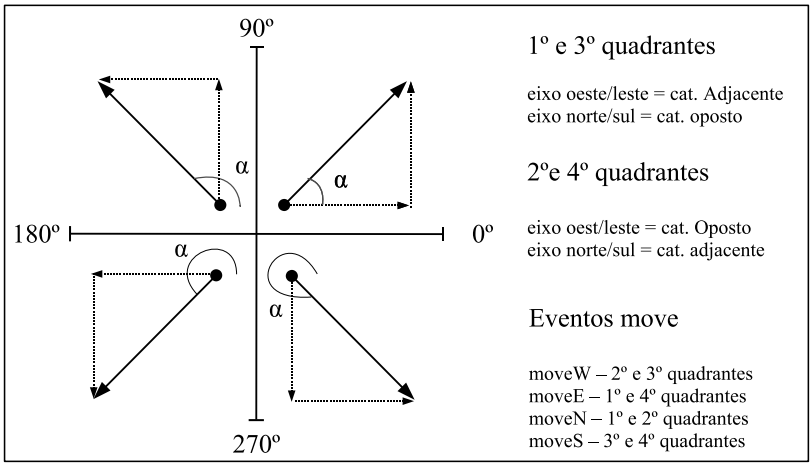


Figura 57 – Identificar catetos oposto e adjacente, pelo quadrante, para possibilitar a execução do movimento.

Identificados os eventos $move$ que devem ocorrer e o tamanho do movimento em cada um dos eixos, os eventos $move$ são setados de acordo com o velocidade necessária para o nodo executar o movimento (os catetos do triângulo retângulo, que tem como hipotenusa o movimento do nodo). O movimento será uma combinação de eventos $moveW$ ou $moveE$ e $moveN$ ou $moveS$. Desta forma, é gerada a SAN que representa a modelagem completa do padrão de movimento *Random Direction* em área 2D, com pode ser visto na Figura 58.

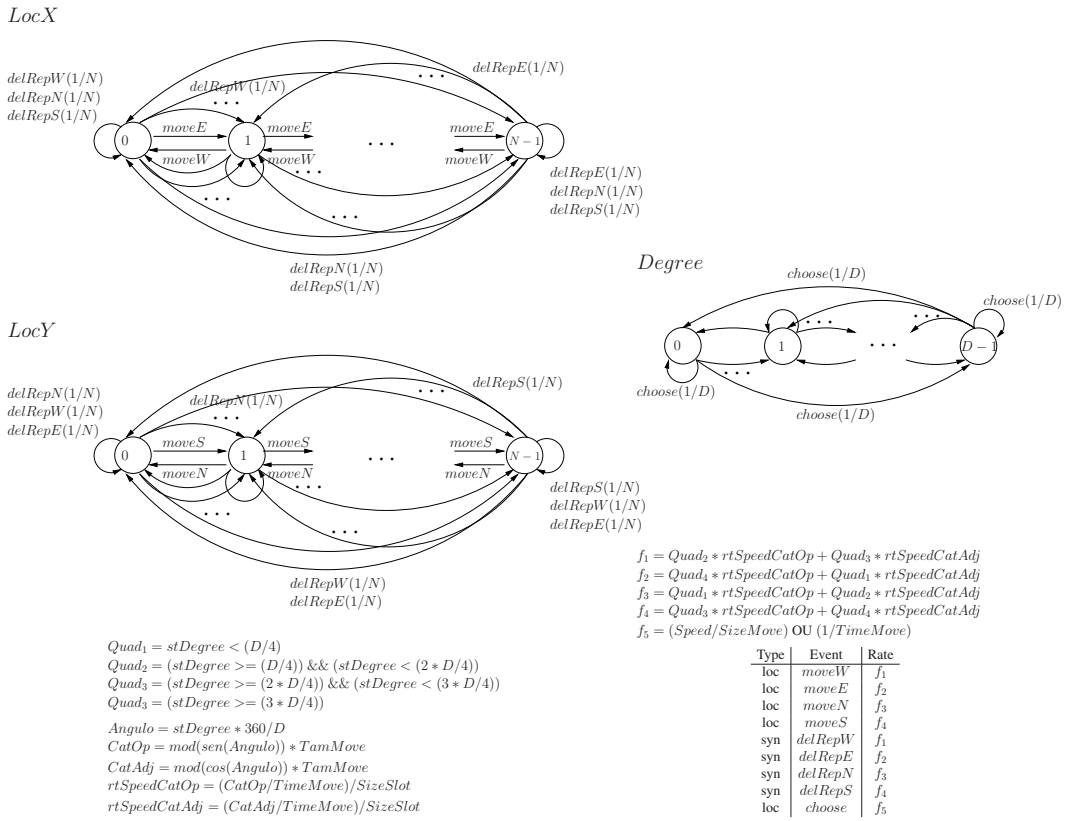


Figura 58 – Random Direction com *Delete and Replace* em área 2D

5.3.2 Resultados para Modelo em 2D

Na Figura 59 são demonstrados graficamente alguns resultados obtidos para uma área de movimentação de $1000\ m^2$, discretizada em $20 \times 20\ slots$, considerando variação do tamanho do movimento em 100, 500, 1000 e 5000 metros. Importante lembrar que, a velocidade e tempo de pausa são irrelevantes para os resultados, uma vez que velocidade e pausa não influenciam no resultado obtido. Esses resultados apresentados graficamente demonstram um resultado consistente quando comparados com os resultados de [3].

Observando os resultados apresentados na Figura 59 pode-se perceber que, assim como no modelo 1D, quando decrementado o tamanho do movimento o nodo tende a concentrar-se na região central da área de movimentação, analogamente, aumentando o tamanho do movimento a distribuição espacial tende a ser uniforme.

É importante relembrar, que no modelo 2D foi assumida a discretização do número de possíveis direções para o movimento. Figura 60 demonstra a variação de acordo com um menor ou maior número de direções. Uma análise sobre estes resultados leva a seguinte conclusão: o número de direções modeladas tem um impacto sobre os resultados deste modelo, quanto maior o número de direções, maior a precisão dos resultados. Porém, o impacto pode ser pouco relevante.

O gráfico da Figura 61 demonstra a diferença de resultados conforme o aumento no número

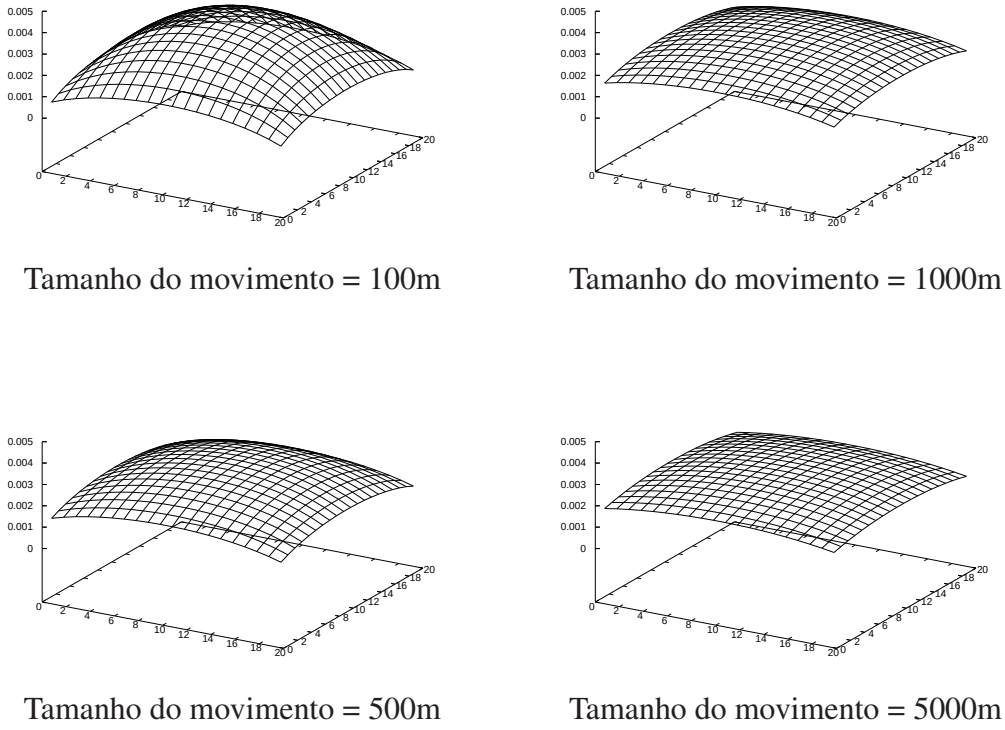
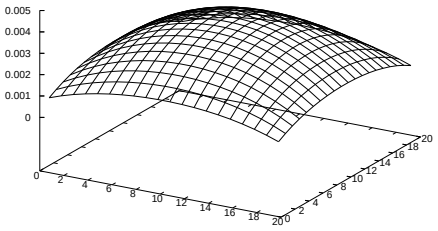


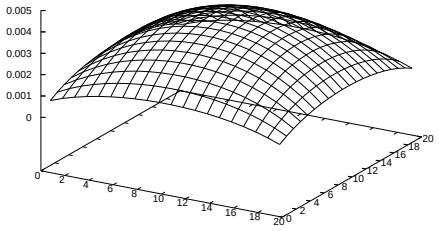
Figura 59 – Distribuição espacial com a variação do tamanho do movimento em: 100m, 500m, 1000m, 5000m

de direções possíveis. No gráfico, a curva no ponto $X = 4$ representa a diferença de resultados obtidos entre 4 e 8 graus; a curva no ponto $X = 16$ representa a diferença entre resultados com 16 e 20 graus. Desta forma, pode-se observar que a utilização de mais de 12 graus passa a ser pouco significativo nos resultados obtidos. Em compensação, o aumento do número de graus impacta no tamanho do espaço de estados do modelo SAN, o que pode não ser interessante.

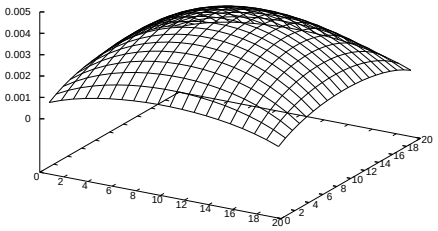
Além de estudar o efeito da variação de pausa e da velocidade no movimento, também foi analisado o efeito da decisão tomada pelo nodo quando efetuar o *Replace*. Foram consideradas duas possibilidades: o nodo continua o movimento que estava executando ao incidir sobre a borda, ou o nodo inicia novo movimento, escolhendo nova direção. No modelo com uma dimensão, esta decisão não causou efeito nos resultados, porém no modelo em duas dimensões foi encontrada uma diferença pouco relevante e que diminui com o aumento do número de estados do autômato $Degree$.



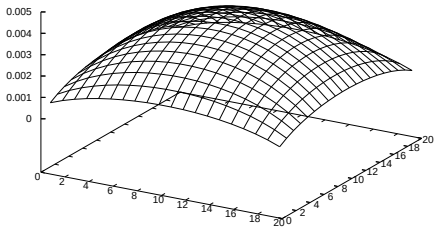
4 graus



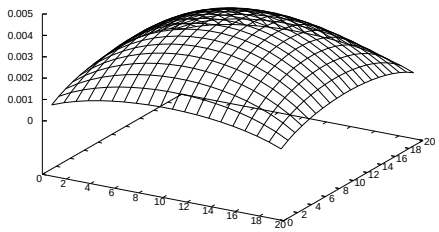
8 graus



12 graus



16 graus



20 graus

Figura 60 – Variação do número de estados do autômato Degree: 4, 8, 12, 16 e 20 estados (graus).

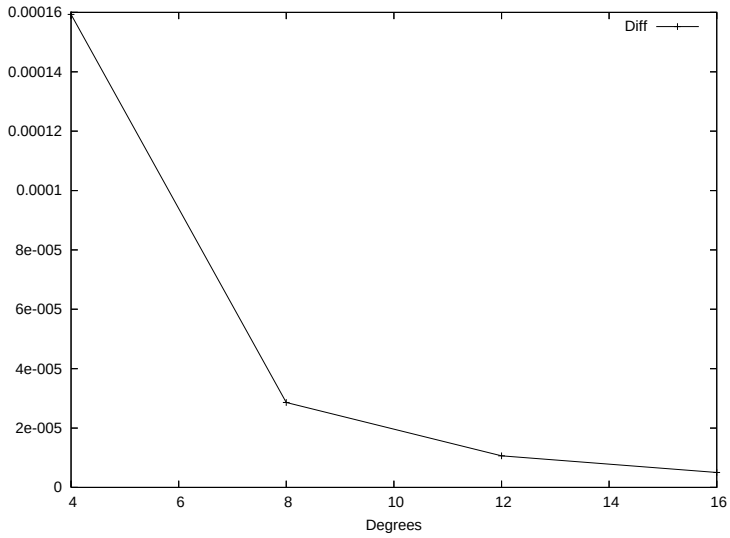


Figura 61 – Impacto do aumento do número de graus nos resultados obtidos.

6 Conclusões e Trabalhos Futuros

Nesta dissertação, foram apresentados modelos de representação para o padrão de movimentação *Random Direction* e *Random Waypoint*, bem como as diferentes características de cada um, utilizando o formalismo SAN, de forma a obter uma análise estacionária de localização de dispositivos móveis.

Os resultados obtidos para o padrão RWP foram comparados com outros resultados encontrados na literatura, baseados em simulação e métodos analíticos, obtendo-se compatibilidade numérica e de comportamento entre eles. A comparação numérica para o padrão RD não foi possível, pois não foram encontrados trabalhos indicando resultados precisos. Porém, a análise baseou-se na comparação visual de gráficos, comparando as diferentes variações dos modelos (tipos de bordas e variações de pausa e velocidade).

A utilização de um formalismo markoviano estruturado, as Redes de Autômatos Estocásticos, possibilitou grande flexibilidade e facilidade no momento de estender os modelos. De certa maneira, foi criado um modelo básico para representar a movimentação (a localização dos dispositivos na área de movimentação) do nodo e, a partir deste, foram modelados novos comportamentos, representando as características de cada variação dos padrões de movimentação.

A principal limitação prevista neste trabalho era o problema da explosão do espaço de estados. O número de estados dos modelos está diretamente relacionado com a discretização da escolha de direção no modelo para RD e discretização da área de movimentação dos modelos para RWP e RD, impactando na precisão dos resultados obtidos. Entretanto, os números de estados utilizados nos modelos demonstraram-se suficientes para obter um bom nível de precisão, em relação outros modelos e resultados encontrados na literatura. Portanto, o tamanho do espaço de estados não foi um fator limitante deste trabalho. Porém, estender estes modelos para representar novos comportamentos irá requerer cuidados quanto ao aumento no espaço de estados.

A contribuição deste trabalho refere-se a um entendimento detalhado dos padrões de movimentação de escolha randômica (RD e RWP), com suas possíveis variações: comportamento de bordas, velocidades, tempo de pausa e definições de caminhos. Outra contribuição é da utilização de um formalismo markoviano estruturado, que possibilita estender os modelos apresentados de forma a representar novas características e, até mesmo, analisar outras métricas, diferentes da distribuição espacial de nodos, que poderiam ser usadas na avaliação de desempenho de redes *Ad Hoc*.

Finalmente, a sugestão de trabalhos futuros trata da continuidade do estudo de padrões de movimentação, de maneira a estender os modelos propostos para representar diversas outras

realidades, como áreas de movimentação em três dimensões e obstáculos na área de movimentação. Nesta sugestão, enquadram-se padrões de movimentação para áreas urbanas (*City Section*), para estradas (*Freeway*) e padrões para movimentação de grupos.

Ainda na continuidade dos estudos sobre os padrões de movimentação, também é importante analisar a relação do padrão de movimentação e a conectividade, uma vez que a distribuição espacial não indica diretamente o nível de conectividade da rede. Para isto, deverá ser feito um estudo da relação da distribuição espacial, densidade da rede e relação temporal dos nodos, obtendo, possivelmente, métricas como: duração de enlaces e duração de caminhos.

Considerando a possibilidade de análise de métricas de conectividade, uma extensão natural é a análise de desempenho de protocolos de roteamento para redes *Ad Hoc*, possibilitando avaliações de comunicação fim-a-fim. As avaliações de comunicação fim-a-fim ou de desempenho de protocolos de roteamento são normalmente efetuadas através de simulação. Neste sentido, a sugestão é de efetuar estas análises utilizando métodos analíticos, possibilitando, até certo ponto, a validação de modelos e, para alguns casos, podendo aumentar a acuracidade das previsões.

Referências

[1] F. Bai, N. Sadagopan, and A. Helmy. The important framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks. *Elsevier Ad Hoc Networks*, 1(4):383–403, November 2003.

[2] A. Benoit, L. Brenner, P. H. Lemelle, B. Plateau, and W. J. Stewart. The peps software tool. In *Computer Performance Evaluation / TOOLS 2003*, volume 2794 of *LNCS*, pages 98–115, Urbana, IL, USA, 2003. Springer-Verlag Heidelberg.

[3] C. Bettstetter. Mobility modeling in wireless networks: Categorization, smooth movement, and border effect. *ACM SIGMOBILE Mobile Computing and Communications*, 5(3):55–66, July 2001.

[4] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, July - September 2003.

[5] L. Brenner, P. H. Lemelle, and A. Sales. The need for and the advantages of generalized tensor algebra for kronecker structured representations. *International Journal of Simulation: Systems, Science & Technology*, 6(3-4):52–60, February 2005.

[6] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, August 2002.

[7] F. L. Delamare, F. L. Dotti, C. M. Nunes, P. Fernandes, and L. Ost. Analytical modeling of random waypoint mobility patterns. In *ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 106–113, Terromolinos, Espanha, October 2006.

[8] Z. J. Haas. A new protocol for the reconfigurable wireless networks. In *IEEE International Conference on Universal Personal Communication (ICUPC)*, pages 562–565, October 1997.

[9] Z. J. Haas and M. R. Pearlman. The performance of query control schemes for the zone routing protocol. In *ACM SIGCOMM*, pages 167–177, Vancover, Canadá, September 1998.

[10] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hocwireless networks. In *2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, August 1999.

[11] X. Hong, T. J. Kwon, M. Gerla, D. L. Gu, and G. Pei. A mobility framework for ad hoc wireless networks. In *ACM Second International Conference on Mobile Data Management (MDM '2001)*, volume 1987, pages 185–196, London, UK, January 2001. Springer-Verlag.

[12] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[13] T. J. Kwon and M. Gerla. Clustering with power control. In *IEEE Military Communications (MILCOM'99)*, volume 2, pages 1424–1428, 1999.

[14] B. Liang and Z. J. Haas. Predictive distance-based mobility management for pcs networks. In *Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, NY, March 1999.

[15] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, January - February 2004.

[16] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the imapct of alternate path routing for load balacing in mobile ad hoc networks. In *Workshop on Mobile Ad Hoc Network and Computing (MobiHOV)*, Boston, USA, August 2000.

[17] B. Plateau and K. Atif. Stochastic automata networks for modelling parallel systems. *IEEE Transactions on Software Enginneering*, 17(10):1093–1108, October 1991.

[18] G. Resta and P. Santi. An analysis of the node spatial distribuition of random waypoint mobility model for ad hoc networks. In *ACM International Workshop on Performance Evaluation of Wireless Ad Hoc*, pages 44–50, Toulouse, France, 2002.

[19] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *IEEE International Conference on Communications*, volume 3, pages 857–861, June 2001.

[20] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[21] UCLA Parallel Computing Laboratory - University of California. *Global Mobile Information System Simulation Library - GloMoSim.* Capturado em: http://pcl.cs.ucla.edu/projects/glomosim/, Dezembro 2006.

[22] The VINT PROJECT. *The Network Simulator ns-2.* Capturado em: http://nsnam.isi.edu/nsnam/index.php, Dezembro 2006.

[23] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Ann. Joint Conf. IEEE Computer and Comm. Soc. (INFOCOM 2003)*, volume 2, pages 1312–1321, April 2003.

[24] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *ACM/IEEE Int'l Conf. Mobile Computing and Networking (MOBICOM '03)*, pages 205–216, New York, NY, USA, 2003. ACM Press.

Apêndice A – SAN para RWP em 1D

SAN da Figura 25, modelo para padrão *Random Waypoint* em uma dimensão.

```
identifiers

    Velocidade1  = 20 ; // Velocidade (m/s)
    Parada1      = 0.001 ; // Pausa (s)
    NumEspacos   = 20 ;
    Esp          = [0..19] ;
    TamArea      = 1000 ; // tamanho da area de movimentacao (m)
    TamEspaco    = (TamArea / NumEspacos) ; // Tamanho do slot (m)
    TaxaVel1     = Velocidade1 / TamEspaco ;  // Taxa de mudanca de localizacao
    TaxaDecDest1 = ((st Dest_MN1)==(st Loc_MN1))*(1/(Parada1*(NumEspacos -1)));
    TaxaAndarDir1 = TaxaVel1 * ( st Dest_MN1 > st Loc_MN1 ) ;
    TaxaAndarEsq1 = TaxaVel1 * ( st Dest_MN1 < st Loc_MN1 ) ;

events

    loc txm1(TaxaDecDest1);
    loc txAndarD1(TaxaAndarDir1);
    loc txAndarE1(TaxaAndarEsq1);

reachability = 1;

network mobilidade(continuous)

    aut Dest_MN1
        stt Dest0
            to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1 to(Dest4) txm1
            to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
            to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
            to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest1
            to(Dest0) txm1 to(Dest2) txm1 to(Dest3) txm1 to(Dest4) txm1
            to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
            to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
            to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest2
            to(Dest0) txm1 to(Dest1) txm1 to(Dest3) txm1 to(Dest4) txm1
            to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
            to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
            to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest3
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest4) txm1
            to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
            to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
            to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
```

```
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest4
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
    to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest5
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest6) txm1 to(Dest7) txm1 to(Dest8) txm1
    to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest6
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest7) txm1 to(Dest8) txm1
    to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest7
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest8) txm1
    to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest8
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
    to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest9
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
    to(Dest8) txm1 to(Dest10) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest10
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
    to(Dest8) txm1 to(Dest9) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest11
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
    to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest12
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
    to(Dest8) txm1 to(Dest9) txm1 to(Dest11) txm1 to(Dest12) txm1
    to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
    to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
stt Dest13
    to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
    to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
```

```
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest14) txm1 to(Dest15) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest14
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest15) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest15
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest14) txm1 to(Dest16) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest16
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1
            to(Dest17) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest17
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1
            to(Dest16) txm1 to(Dest18) txm1 to(Dest19) txm1
        stt Dest18
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1
            to(Dest16) txm1 to(Dest17) txm1 to(Dest19) txm1
        stt Dest19
            to(Dest0) txm1 to(Dest1) txm1 to(Dest2) txm1 to(Dest3) txm1
            to(Dest4) txm1 to(Dest5) txm1 to(Dest6) txm1 to(Dest7) txm1
            to(Dest8) txm1 to(Dest9) txm1 to(Dest10) txm1 to(Dest11) txm1
            to(Dest12) txm1 to(Dest13) txm1 to(Dest14) txm1 to(Dest15) txm1
            to(Dest16) txm1 to(Dest17) txm1 to(Dest18) txm1


aut Loc_MN1
       stt E[Esp]
            to(--) txAndarE1 to(++) txAndarD1

results
    prob_MN1_E0 = (st Loc_MN1 == 0) ;
    prob_MN1_E1 = (st Loc_MN1 == 1) ;
    prob_MN1_E2 = (st Loc_MN1 == 2) ;
    prob_MN1_E3 = (st Loc_MN1 == 3) ;
    prob_MN1_E4 = (st Loc_MN1 == 4) ;
    prob_MN1_E5 = (st Loc_MN1 == 5) ;
    prob_MN1_E6 = (st Loc_MN1 == 6) ;
    prob_MN1_E7 = (st Loc_MN1 == 7) ;
    prob_MN1_E8 = (st Loc_MN1 == 8) ;
    prob_MN1_E9 = (st Loc_MN1 == 9) ;
    prob_MN1_E10 = (st Loc_MN1 == 10) ;
    prob_MN1_E11 = (st Loc_MN1 == 11) ;
```

```
prob_MN1_E12 = (st Loc_MN1 == 12) ;
prob_MN1_E13 = (st Loc_MN1 == 13) ;
prob_MN1_E14 = (st Loc_MN1 == 14) ;
prob_MN1_E15 = (st Loc_MN1 == 15) ;
prob_MN1_E16 = (st Loc_MN1 == 16) ;
prob_MN1_E17 = (st Loc_MN1 == 17) ;
prob_MN1_E18 = (st Loc_MN1 == 18) ;
prob_MN1_E19 = (st Loc_MN1 == 19) ;
```

Apêndice B – SAN para RWP em 2D

SAN da Figura 29, modelo para padrão *Random Waypoint* em duas dimensões.

```
identifiers

    Velocidade1   = 20 ; // velocidade do nodo (m/s)
    Parada1       = 0.001 ; // Pausa (s)
    NumEspacos    = 20 ;
    Esp           = [0..19] ;
    TamArea       = 1000 ;
    TamEspaco     = (TamArea / NumEspacos) ;
    TaxaVel1      = Velocidade1 / TamEspaco ;

    TaxaDecDest1  = ( ( (st DestX_MN1) == (st LocX_MN1) ) &&
                      ( (st DestY_MN1) == (st LocY_MN1) ) ) *
                      (1 / (Parada1 * (NumEspacos * NumEspacos - 1)  ) );

    TaxaAndarL1   = TaxaVel1 * ( ( (st DestX_MN1) > (st LocX_MN1) ) &&
            ( ((st DestX_MN1) - (st LocX_MN1)) >=
            (max((st DestY_MN1)-(st LocY_MN1),(st LocY_MN1)-(st DestY_MN1)))));

    TaxaAndarO1   = TaxaVel1 * ( ( (st DestX_MN1) < (st LocX_MN1) ) &&
            ( ((st LocX_MN1) - (st DestX_MN1)) >=
            (max((st DestY_MN1)-(st LocY_MN1),(st LocY_MN1)-(st DestY_MN1)))));

    TaxaAndarN1   = TaxaVel1 * ( ( (st DestY_MN1) > (st LocY_MN1) ) &&
            ( ((st DestY_MN1) - (st LocY_MN1)) >=
            (max((st DestX_MN1)-(st LocX_MN1),(st LocX_MN1)-(st DestX_MN1)))));

    TaxaAndarS1   = TaxaVel1 * ( ( (st DestY_MN1) < (st LocY_MN1) )  &&
            ( ((st LocY_MN1) - (st DestY_MN1)) >=
            (max((st DestX_MN1)-(st LocX_MN1),(st LocX_MN1)-(st DestX_MN1)))));

events

    loc txAndarL1(TaxaAndarL1);
    loc txAndarO1(TaxaAndarO1);
    loc txAndarN1(TaxaAndarN1);
    loc txAndarS1(TaxaAndarS1);
    loc txndVH1(TaxaDecDest1); // movimento na Vertical ou horizontal
    syn txnd1(TaxaDecDest1); // Outros movimentos

reachability = 1;

network mobilidade(continuous)

    aut DestX_MN1
        stt Dest0
            to(Dest1) txnd1 txndVH1 to(Dest2) txnd1 txndVH1
            to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
```

```
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest1
    to(Dest0) txnd1 txndVH1 to(Dest2) txnd1 txndVH1
    to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest2
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest3
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest4
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest5
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
```

```
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest6
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest7
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest8
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest9
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest10
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
   to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
   to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
   to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
```

```
   to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
   to(Dest19) txnd1 txndVH1
stt Dest11
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest12
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest13
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest14
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest15
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
    to(Dest14) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest16
```

```
        to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
        to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
        to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
        to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
        to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
        to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
        to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
        to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
        to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
        to(Dest19) txnd1 txndVH1
    stt Dest17
        to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
        to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
        to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
        to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
        to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
        to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
        to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
        to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
        to(Dest16) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
        to(Dest19) txnd1 txndVH1
    stt Dest18
        to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
        to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
        to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
        to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
        to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
        to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
        to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
        to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
        to(Dest16) txnd1 txndVH1 to(Dest17) txnd1 txndVH1
        to(Dest19) txnd1 txndVH1
    stt Dest19
        to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
        to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
        to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
        to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
        to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
        to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
        to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
        to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
        to(Dest16) txnd1 txndVH1 to(Dest17) txnd1 txndVH1
        to(Dest18) txnd1 txndVH1

aut LocX_MN1
    stt E[Esp]
        to(--) txAndarO1 to(++) txAndarL1

aut DestY_MN1
    stt Dest0
        to(Dest1) txnd1 txndVH1 to(Dest2) txnd1 txndVH1
        to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
        to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
        to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
        to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
        to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
        to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
        to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
```

```
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest1
    to(Dest0) txnd1 txndVH1 to(Dest2) txnd1 txndVH1
    to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest2
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest3) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest3
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest4) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest4
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest5) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest5
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest6) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest6
```

```
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest7) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest7
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest8) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest8
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest9) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest9
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest10) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest10
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest11) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest11
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
```

```
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest12) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest12
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest13) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest13
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest14) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest14
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
    to(Dest15) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest15
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
    to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
    to(Dest14) txnd1 txndVH1 to(Dest16) txnd1 txndVH1
    to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
    to(Dest19) txnd1 txndVH1
stt Dest16
    to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
    to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
    to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
    to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
    to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
    to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
```

```
            to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
            to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
            to(Dest17) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
            to(Dest19) txnd1 txndVH1
        stt Dest17
            to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
            to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
            to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
            to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
            to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
            to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
            to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
            to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
            to(Dest16) txnd1 txndVH1 to(Dest18) txnd1 txndVH1
            to(Dest19) txnd1 txndVH1
        stt Dest18
            to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
            to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
            to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
            to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
            to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
            to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
            to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
            to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
            to(Dest16) txnd1 txndVH1 to(Dest17) txnd1 txndVH1
            to(Dest19) txnd1 txndVH1
        stt Dest19
            to(Dest0) txnd1 txndVH1 to(Dest1) txnd1 txndVH1
            to(Dest2) txnd1 txndVH1 to(Dest3) txnd1 txndVH1
            to(Dest4) txnd1 txndVH1 to(Dest5) txnd1 txndVH1
            to(Dest6) txnd1 txndVH1 to(Dest7) txnd1 txndVH1
            to(Dest8) txnd1 txndVH1 to(Dest9) txnd1 txndVH1
            to(Dest10) txnd1 txndVH1 to(Dest11) txnd1 txndVH1
            to(Dest12) txnd1 txndVH1 to(Dest13) txnd1 txndVH1
            to(Dest14) txnd1 txndVH1 to(Dest15) txnd1 txndVH1
            to(Dest16) txnd1 txndVH1 to(Dest17) txnd1 txndVH1
            to(Dest18) txnd1 txndVH1

    aut LocY_MN1
        stt E[Esp]
            to(--) txAndarS1 to(++) txAndarN1

results
    prob_MN1_E_00_00  = (st LocX_MN1 == 00) && (st LocY_MN1 == 00) ;
    prob_MN1_E_00_01  = (st LocX_MN1 == 00) && (st LocY_MN1 == 01) ;
    prob_MN1_E_00_02  = (st LocX_MN1 == 00) && (st LocY_MN1 == 02) ;
    prob_MN1_E_00_03  = (st LocX_MN1 == 00) && (st LocY_MN1 == 03) ;
    prob_MN1_E_00_04  = (st LocX_MN1 == 00) && (st LocY_MN1 == 04) ;
    prob_MN1_E_00_05  = (st LocX_MN1 == 00) && (st LocY_MN1 == 05) ;
    prob_MN1_E_00_06  = (st LocX_MN1 == 00) && (st LocY_MN1 == 06) ;
    prob_MN1_E_00_07  = (st LocX_MN1 == 00) && (st LocY_MN1 == 07) ;
    ...
    prob_MN1_E_19_15  = (st LocX_MN1 == 19) && (st LocY_MN1 == 15) ;
    prob_MN1_E_19_16  = (st LocX_MN1 == 19) && (st LocY_MN1 == 16) ;
    prob_MN1_E_19_17  = (st LocX_MN1 == 19) && (st LocY_MN1 == 17) ;
    prob_MN1_E_19_18  = (st LocX_MN1 == 19) && (st LocY_MN1 == 18) ;
    prob_MN1_E_19_19  = (st LocX_MN1 == 19) && (st LocY_MN1 == 19) ;
```

Apêndice C – SAN para padrão RD de Royer em 1D

Descrição textual da SAN da Figura 35, padrão RD de Royer sem pausa e em área de uma dimensão.

```
identifiers

    Speed       = 4 ;
    NumSlots    = 20 ;
    Pause       = 0 ;
    SizeArea    = 1000 ;
    SizeSlot    = (SizeArea / NumSlots) ;
    rtSpeed     = ( Speed / SizeSlot ) ;
    rtMoveW     = (st DirX == W) * ( rtSpeed ) ;
    rtMoveE     = (st DirX == E) * ( rtSpeed ) ;

events

    syn chooseE(rtMoveW);
    syn chooseW(rtMoveE);
    loc moveW(rtMoveW);
    loc moveE(rtMoveE);

reachability = 1 ;


network rd(continuous)

    aut LocX
        stt S0
            to(S0) chooseE to(S1) moveE
        stt S1
            to(S0) moveW to(S2) moveE
        stt S2
            to(S1) moveW to(S3) moveE
        stt S3
            to(S2) moveW to(S4) moveE
        stt S4
            to(S3) moveW to(S5) moveE
        stt S5
            to(S4) moveW to(S6) moveE
        stt S6
            to(S5) moveW to(S7) moveE
        stt S7
            to(S6) moveW to(S8) moveE
        stt S8
            to(S7) moveW to(S9) moveE
        stt S9
            to(S8) moveW to(S10) moveE
        stt S10
            to(S9) moveW to(S11) moveE
```

```
        stt S11
            to(S10) moveW to(S12) moveE
        stt S12
            to(S11) moveW to(S13) moveE
        stt S13
            to(S12) moveW to(S14) moveE
        stt S14
            to(S13) moveW to(S15) moveE
        stt S15
            to(S14) moveW to(S16) moveE
        stt S16
            to(S15) moveW to(S17) moveE
        stt S17
            to(S16) moveW to(S18) moveE
        stt S18
            to(S17) moveW to(S19) moveE
        stt S19
            to(S18) moveW to(S19) chooseW

    aut DirX
        stt W
            to(E) chooseE
        stt E
            to(W) chooseW

results
    prob_MN1_E0 = (st LocX == 0) ;
    prob_MN1_E1 = (st LocX == 1) ;
    prob_MN1_E2 = (st LocX == 2) ;
    prob_MN1_E3 = (st LocX == 3) ;
    prob_MN1_E4 = (st LocX == 4) ;
    prob_MN1_E5 = (st LocX == 5) ;
    prob_MN1_E6 = (st LocX == 6) ;
    prob_MN1_E7 = (st LocX == 7) ;
    prob_MN1_E8 = (st LocX == 8) ;
    prob_MN1_E9 = (st LocX == 9) ;
    prob_MN1_E10 = (st LocX == 10) ;
    prob_MN1_E11 = (st LocX == 11) ;
    prob_MN1_E12 = (st LocX == 12) ;
    prob_MN1_E13 = (st LocX == 13) ;
    prob_MN1_E14 = (st LocX == 14) ;
    prob_MN1_E15 = (st LocX == 15) ;
    prob_MN1_E16 = (st LocX == 16) ;
    prob_MN1_E17 = (st LocX == 17) ;
    prob_MN1_E18 = (st LocX == 18) ;
    prob_MN1_E19 = (st LocX == 19) ;
```

Apêndice D – SAN para RD - Delete and Replace em 1D

Descrição textual da SAN da Figura 39, padrão RD com regra de borda *Delete and Replace* em área de uma dimensão.

```
identifiers

    Speed        = 32 ;
    Pause        = 32 ;
    DistanceMov  = 200 ;
    TimeMo       = 0 ;
    NumSlots     = 20 ;
    SizeArea     = 1000 ;

    SizeSlot     = (SizeArea / NumSlots) ;
    rtSpeed      = ( Speed / SizeSlot ) ;
    rtChooseX    = ( Speed / DistanceMov ) ;

    rtIniMov     = (1 / ( Pause  ) ) ;
    rtMoveW      = (st DirX == W) * ( rtSpeed ) ;
    rtMoveE      = (st DirX == E) * ( rtSpeed ) ;

    rtDelRepW = ( rtMoveW / NumSlots ) ;
    rtDelRepE = ( rtMoveE / NumSlots ) ;

events

    loc chooseX(rtChooseX);
    loc iniMov(rtIniMov) ;
    loc moveW(rtMoveW);
    loc moveE(rtMoveE);

    syn delRepW(rtDelRepW) ;
    syn delRepE(rtDelRepE) ;

reachability = 1 ;

network rd(continuous)

aut LocX
    stt S0
        to(S0) delRepW to(S1) moveE delRepW to(S2) delRepW to(S3) delRepW
        to(S4) delRepW to(S5) delRepW to(S6) delRepW to(S7) delRepW
        to(S8) delRepW to(S9) delRepW to(S10) delRepW to(S11) delRepW
        to(S12) delRepW to(S13) delRepW to(S14) delRepW to(S15) delRepW
        to(S16) delRepW to(S17) delRepW to(S18) delRepW to(S19) delRepW
    stt S1
        to(S0) moveW to(S2) moveE
    stt S2
        to(S1) moveW to(S3) moveE
    stt S3
```

```
        to(S2) moveW to(S4) moveE
    stt S4
        to(S3) moveW to(S5) moveE
    stt S5
        to(S4) moveW to(S6) moveE
    stt S6
        to(S5) moveW to(S7) moveE
    stt S7
        to(S6) moveW to(S8) moveE
    stt S8
        to(S7) moveW to(S9) moveE
    stt S9
        to(S8) moveW to(S10) moveE
    stt S10
        to(S9) moveW to(S11) moveE
    stt S11
        to(S10) moveW to(S12) moveE
    stt S12
        to(S11) moveW to(S13) moveE
    stt S13
        to(S12) moveW to(S14) moveE
    stt S14
        to(S13) moveW to(S15) moveE
    stt S15
        to(S14) moveW to(S16) moveE
    stt S16
         to(S15) moveW to(S17) moveE
    stt S17
         to(S16) moveW to(S18) moveE
    stt S18
         to(S17) moveW to(S19) moveE
    stt S19
         to(S0) delRepE to(S1) delRepE to(S2) delRepE to(S3) delRepE
         to(S4) delRepE to(S5) delRepE to(S6) delRepE to(S7) delRepE
         to(S8) delRepE to(S9) delRepE  to(S10) delRepE to(S11) delRepE
         to(S12) delRepE to(S13) delRepE to(S14) delRepE to(S15) delRepE
         to(S16) delRepE to(S17) delRepE to(S18) delRepE moveW to(S19) delRepE


aut DirX
    stt W
         to(P) chooseX to(E) delRepW
    stt P
         to(E) iniMov(0.5) to(W) iniMov(0.5)
    stt E
         to(P) chooseX to(W) delRepE

results
    prob_MN1_E0 = (st LocX == 0) ;
    prob_MN1_E1 = (st LocX == 1) ;
    prob_MN1_E2 = (st LocX == 2) ;
    prob_MN1_E3 = (st LocX == 3) ;
    prob_MN1_E4 = (st LocX == 4) ;
    prob_MN1_E5 = (st LocX == 5) ;
    prob_MN1_E6 = (st LocX == 6) ;
    prob_MN1_E7 = (st LocX == 7) ;
    prob_MN1_E8 = (st LocX == 8) ;
    prob_MN1_E9 = (st LocX == 9) ;
```

```
prob_MN1_E10 = (st LocX == 10) ;
prob_MN1_E11 = (st LocX == 11) ;
prob_MN1_E12 = (st LocX == 12) ;
prob_MN1_E13 = (st LocX == 13) ;
prob_MN1_E14 = (st LocX == 14) ;
prob_MN1_E15 = (st LocX == 15) ;
prob_MN1_E16 = (st LocX == 16) ;
prob_MN1_E17 = (st LocX == 17) ;
prob_MN1_E18 = (st LocX == 18) ;
prob_MN1_E19 = (st LocX == 19) ;
```

Apêndice E – SAN para RD - Delete and Replace em 2D

Descrição textual da SAN da Figura 58, padrão RD com regra de borda *Delete and Replace* em área de duas dimensões.

```
identifiers

    Speed        = 5 ;
    SizeMov      = 100 ;
    NumG         = 16 ;

    Degrees      = [0..15];
    probCDeg     = ( 1 / NumG ) ;

    NumSlot      = 20 ;
    SizeArea     = 1000 ;

    SizeSlot     =  ( SizeArea  /  NumSlot ) ;
    TimeMov      =  ( SizeMov / Speed ) ;

    SizeMoveW    = (( (st Degree == 8) * (SizeMov) ) +
        (((st Degree == 9) || (st Degree == 7) ) *
          (SizeMov * 0.92387953251 ) ) + ( ( (st Degree == 10) ||
          (st Degree == 6) ) * (SizeMov * 0.70710678118 ) ) +
         (((st Degree==11) || (st Degree==5)) * (SizeMov * 0.38268343236)));

    SizeMoveE    = (( (st Degree == 0) * (SizeMov) ) +
        (((st Degree == 1) || (st Degree == 15) ) *
          (SizeMov * 0.92387953251 ) ) + ( ( (st Degree == 2) ||
          (st Degree == 14) ) * (SizeMov * 0.70710678118 ) ) +
         (((st Degree==3) || (st Degree==13) ) * (SizeMov * 0.38268343236)));

    SizeMoveN    = (( ( (st Degree == 1) || (st Degree == 7) ) *
        (SizeMov * 0.38268343236)) + ( ( (st Degree == 2) ||
        (st Degree == 6) ) * (SizeMov * 0.70710678118)) +
        (((st Degree == 3) || (st Degree == 5)) * (SizeMov * 0.92387953251)) +
        ((st Degree == 4) * (SizeMov))) ;

    SizeMoveS    = (( ( (st Degree == 9) || (st Degree == 15) ) *
        (SizeMov * 0.38268343236)) + ( ( (st Degree == 10) ||
        (st Degree == 14) ) * (SizeMov * 0.70710678118)) +
        (((st Degree == 11) || (st Degree == 13)) * (SizeMov * 0.92387953251))+
        ((st Degree == 12) * (SizeMov))) ;

    SpeedW = ( SizeMoveW / TimeMov ) ;
    SpeedE = ( SizeMoveE / TimeMov ) ;
    SpeedN = ( SizeMoveN / TimeMov ) ;
    SpeedS = ( SizeMoveS / TimeMov ) ;
```

```
    rtMoveW = ( ( SpeedW / SizeSlot ) ) ;
    rtMoveE = ( ( SpeedE / SizeSlot ) ) ;
    rtMoveN = ( ( SpeedN / SizeSlot ) ) ;
    rtMoveS = ( ( SpeedS / SizeSlot ) ) ;

    rtChoose = (  1 / TimeMov ) ;

    rtDelRepW = (rtMoveW ) ;
    rtDelRepE = (rtMoveE ) ;
    rtDelRepN = (rtMoveN ) ;
    rtDelRepS = (rtMoveS ) ;

events

loc choose(rtChoose);

syn delRepW(rtDelRepW);
syn delRepE(rtDelRepE);
syn delRepN(rtDelRepN);
syn delRepS(rtDelRepS);

loc moveW(rtMoveW);
loc moveE(rtMoveE);
loc moveN(rtMoveN);
loc moveS(rtMoveS);


reachability = 1;

network rd(continuous)

    aut Degree
        stt D0
            to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
                delRepN(probCDeg) delRepS(probCDeg)
            to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
```

```
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D1
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D2
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
```

```
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D3
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D4
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
```

```
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D5
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D6
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
```

```
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D7
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
```

```
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D8
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D9
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
```

```
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D10
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D11
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
```

```
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D12
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
    to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
stt D13
    to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
        delRepN(probCDeg) delRepS(probCDeg)
```

```
        to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
    stt D14
        to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
```

```
            delRepN(probCDeg) delRepS(probCDeg)
        to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
    stt D15
        to(D0) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D1) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D2) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D3) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D4) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D5) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D6) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D7) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D8) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D9) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D10) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D11) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D12) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D13) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D14) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)
        to(D15) choose(probCDeg) delRepW(probCDeg) delRepE(probCDeg)
            delRepN(probCDeg) delRepS(probCDeg)

aut LocX
    stt S0
        to(S0) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S1) delRepW(0.05) delRepN(0.05) delRepS(0.05) moveE
        to(S2) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S3) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S4) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S5) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S6) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S7) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S8) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S9) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S10) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S11) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S12) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S13) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S14) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S15) delRepW(0.05) delRepN(0.05) delRepS(0.05)
        to(S16) delRepW(0.05) delRepN(0.05) delRepS(0.05)
```

```
    to(S17) delRepW(0.05) delRepN(0.05) delRepS(0.05)
    to(S18) delRepW(0.05) delRepN(0.05) delRepS(0.05)
    to(S19) delRepW(0.05) delRepN(0.05) delRepS(0.05)
stt S1
    to(S0) delRepN(0.05) delRepS(0.05) moveW
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05) moveE
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S2
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05) moveW
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05) moveE
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S3
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05) moveW
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05) moveE
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
```

```
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S4
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05) moveW
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05) moveE
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S5
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05) moveW
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05) moveE
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S6
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05) moveW
    to(S6) delRepN(0.05) delRepS(0.05)
```

```
        to(S7) delRepN(0.05) delRepS(0.05) moveE
        to(S8) delRepN(0.05) delRepS(0.05)
        to(S9) delRepN(0.05) delRepS(0.05)
        to(S10) delRepN(0.05) delRepS(0.05)
        to(S11) delRepN(0.05) delRepS(0.05)
        to(S12) delRepN(0.05) delRepS(0.05)
        to(S13) delRepN(0.05) delRepS(0.05)
        to(S14) delRepN(0.05) delRepS(0.05)
        to(S15) delRepN(0.05) delRepS(0.05)
        to(S16) delRepN(0.05) delRepS(0.05)
        to(S17) delRepN(0.05) delRepS(0.05)
        to(S18) delRepN(0.05) delRepS(0.05)
        to(S19) delRepN(0.05) delRepS(0.05)
    stt S7
        to(S0) delRepN(0.05) delRepS(0.05)
        to(S1) delRepN(0.05) delRepS(0.05)
        to(S2) delRepN(0.05) delRepS(0.05)
        to(S3) delRepN(0.05) delRepS(0.05)
        to(S4) delRepN(0.05) delRepS(0.05)
        to(S5) delRepN(0.05) delRepS(0.05)
        to(S6) delRepN(0.05) delRepS(0.05) moveW
        to(S7) delRepN(0.05) delRepS(0.05)
        to(S8) delRepN(0.05) delRepS(0.05) moveE
        to(S9) delRepN(0.05) delRepS(0.05)
        to(S10) delRepN(0.05) delRepS(0.05)
        to(S11) delRepN(0.05) delRepS(0.05)
        to(S12) delRepN(0.05) delRepS(0.05)
        to(S13) delRepN(0.05) delRepS(0.05)
        to(S14) delRepN(0.05) delRepS(0.05)
        to(S15) delRepN(0.05) delRepS(0.05)
        to(S16) delRepN(0.05) delRepS(0.05)
        to(S17) delRepN(0.05) delRepS(0.05)
        to(S18) delRepN(0.05) delRepS(0.05)
        to(S19) delRepN(0.05) delRepS(0.05)
    stt S8
        to(S0) delRepN(0.05) delRepS(0.05)
        to(S1) delRepN(0.05) delRepS(0.05)
        to(S2) delRepN(0.05) delRepS(0.05)
        to(S3) delRepN(0.05) delRepS(0.05)
        to(S4) delRepN(0.05) delRepS(0.05)
        to(S5) delRepN(0.05) delRepS(0.05)
        to(S6) delRepN(0.05) delRepS(0.05)
        to(S7) delRepN(0.05) delRepS(0.05) moveW
        to(S8) delRepN(0.05) delRepS(0.05)
        to(S9) delRepN(0.05) delRepS(0.05) moveE
        to(S10) delRepN(0.05) delRepS(0.05)
        to(S11) delRepN(0.05) delRepS(0.05)
        to(S12) delRepN(0.05) delRepS(0.05)
        to(S13) delRepN(0.05) delRepS(0.05)
        to(S14) delRepN(0.05) delRepS(0.05)
        to(S15) delRepN(0.05) delRepS(0.05)
        to(S16) delRepN(0.05) delRepS(0.05)
        to(S17) delRepN(0.05) delRepS(0.05)
        to(S18) delRepN(0.05) delRepS(0.05)
        to(S19) delRepN(0.05) delRepS(0.05)
    stt S9
        to(S0) delRepN(0.05) delRepS(0.05)
        to(S1) delRepN(0.05) delRepS(0.05)
```

```
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05) moveW
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05) moveE
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S10
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05) moveW
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05) moveE
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S11
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05) moveW
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05) moveE
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
```

```
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S12
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05) moveW
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05) moveE
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S13
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05) moveW
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05) moveE
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S14
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
```

```
    to(S13) delRepN(0.05) delRepS(0.05) moveW
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05) moveE
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S15
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05) moveW
    to(S15) delRepN(0.05) delRepS(0.05)
    to(S16) delRepN(0.05) delRepS(0.05) moveE
    to(S17) delRepN(0.05) delRepS(0.05)
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S16
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
    to(S8) delRepN(0.05) delRepS(0.05)
    to(S9) delRepN(0.05) delRepS(0.05)
    to(S10) delRepN(0.05) delRepS(0.05)
    to(S11) delRepN(0.05) delRepS(0.05)
    to(S12) delRepN(0.05) delRepS(0.05)
    to(S13) delRepN(0.05) delRepS(0.05)
    to(S14) delRepN(0.05) delRepS(0.05)
    to(S15) delRepN(0.05) delRepS(0.05) moveW
    to(S16) delRepN(0.05) delRepS(0.05)
    to(S17) delRepN(0.05) delRepS(0.05) moveE
    to(S18) delRepN(0.05) delRepS(0.05)
    to(S19) delRepN(0.05) delRepS(0.05)
stt S17
    to(S0) delRepN(0.05) delRepS(0.05)
    to(S1) delRepN(0.05) delRepS(0.05)
    to(S2) delRepN(0.05) delRepS(0.05)
    to(S3) delRepN(0.05) delRepS(0.05)
    to(S4) delRepN(0.05) delRepS(0.05)
    to(S5) delRepN(0.05) delRepS(0.05)
    to(S6) delRepN(0.05) delRepS(0.05)
    to(S7) delRepN(0.05) delRepS(0.05)
```

```
            to(S8) delRepN(0.05) delRepS(0.05)
            to(S9) delRepN(0.05) delRepS(0.05)
            to(S10) delRepN(0.05) delRepS(0.05)
            to(S11) delRepN(0.05) delRepS(0.05)
            to(S12) delRepN(0.05) delRepS(0.05)
            to(S13) delRepN(0.05) delRepS(0.05)
            to(S14) delRepN(0.05) delRepS(0.05)
            to(S15) delRepN(0.05) delRepS(0.05)
            to(S16) delRepN(0.05) delRepS(0.05) moveW
            to(S17) delRepN(0.05) delRepS(0.05)
            to(S18) delRepN(0.05) delRepS(0.05) moveE
            to(S19) delRepN(0.05) delRepS(0.05)
        stt S18
            to(S0) delRepN(0.05) delRepS(0.05)
            to(S1) delRepN(0.05) delRepS(0.05)
            to(S2) delRepN(0.05) delRepS(0.05)
            to(S3) delRepN(0.05) delRepS(0.05)
            to(S4) delRepN(0.05) delRepS(0.05)
            to(S5) delRepN(0.05) delRepS(0.05)
            to(S6) delRepN(0.05) delRepS(0.05)
            to(S7) delRepN(0.05) delRepS(0.05)
            to(S8) delRepN(0.05) delRepS(0.05)
            to(S9) delRepN(0.05) delRepS(0.05)
            to(S10) delRepN(0.05) delRepS(0.05)
            to(S11) delRepN(0.05) delRepS(0.05)
            to(S12) delRepN(0.05) delRepS(0.05)
            to(S13) delRepN(0.05) delRepS(0.05)
            to(S14) delRepN(0.05) delRepS(0.05)
            to(S15) delRepN(0.05) delRepS(0.05)
            to(S16) delRepN(0.05) delRepS(0.05)
            to(S17) delRepN(0.05) delRepS(0.05) moveW
            to(S18) delRepN(0.05) delRepS(0.05)
            to(S19) delRepN(0.05) delRepS(0.05) moveE
        stt S19
            to(S0) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S1) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S2) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S3) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S4) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S5) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S6) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S7) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S8) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S9) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S10) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S11) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S12) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S13) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S14) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S15) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S16) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S17) delRepN(0.05) delRepS(0.05) delRepE(0.05)
            to(S18) delRepN(0.05) delRepS(0.05) delRepE(0.05) moveW
            to(S19) delRepN(0.05) delRepS(0.05) delRepE(0.05)

    aut LocY
        stt S0
            to(S0) delRepW(0.05) delRepE(0.05) delRepN(0.05)
```

```
    to(S1) delRepW(0.05) delRepE(0.05) delRepN(0.05) moveS
    to(S2) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S3) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S4) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S5) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S6) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S7) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S8) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S9) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S10) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S11) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S12) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S13) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S14) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S15) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S16) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S17) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S18) delRepW(0.05) delRepE(0.05) delRepN(0.05)
    to(S19) delRepW(0.05) delRepE(0.05) delRepN(0.05)
stt S1
    to(S0) delRepW(0.05) delRepE(0.05) moveN
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05) moveS
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S2
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05) moveN
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05) moveS
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
```

```
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S3
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05) moveN
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05) moveS
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S4
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05) moveN
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05) moveS
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S5
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05) moveN
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05) moveS
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
```

```
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S6
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05) moveN
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05) moveS
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S7
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05) moveN
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05) moveS
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S8
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
```

```
        to(S7) delRepW(0.05) delRepE(0.05) moveN
        to(S8) delRepW(0.05) delRepE(0.05)
        to(S9) delRepW(0.05) delRepE(0.05) moveS
        to(S10) delRepW(0.05) delRepE(0.05)
        to(S11) delRepW(0.05) delRepE(0.05)
        to(S12) delRepW(0.05) delRepE(0.05)
        to(S13) delRepW(0.05) delRepE(0.05)
        to(S14) delRepW(0.05) delRepE(0.05)
        to(S15) delRepW(0.05) delRepE(0.05)
        to(S16) delRepW(0.05) delRepE(0.05)
        to(S17) delRepW(0.05) delRepE(0.05)
        to(S18) delRepW(0.05) delRepE(0.05)
        to(S19) delRepW(0.05) delRepE(0.05)
    stt S9
        to(S0) delRepW(0.05) delRepE(0.05)
        to(S1) delRepW(0.05) delRepE(0.05)
        to(S2) delRepW(0.05) delRepE(0.05)
        to(S3) delRepW(0.05) delRepE(0.05)
        to(S4) delRepW(0.05) delRepE(0.05)
        to(S5) delRepW(0.05) delRepE(0.05)
        to(S6) delRepW(0.05) delRepE(0.05)
        to(S7) delRepW(0.05) delRepE(0.05)
        to(S8) delRepW(0.05) delRepE(0.05) moveN
        to(S9) delRepW(0.05) delRepE(0.05)
        to(S10) delRepW(0.05) delRepE(0.05) moveS
        to(S11) delRepW(0.05) delRepE(0.05)
        to(S12) delRepW(0.05) delRepE(0.05)
        to(S13) delRepW(0.05) delRepE(0.05)
        to(S14) delRepW(0.05) delRepE(0.05)
        to(S15) delRepW(0.05) delRepE(0.05)
        to(S16) delRepW(0.05) delRepE(0.05)
        to(S17) delRepW(0.05) delRepE(0.05)
        to(S18) delRepW(0.05) delRepE(0.05)
        to(S19) delRepW(0.05) delRepE(0.05)
    stt S10
        to(S0) delRepW(0.05) delRepE(0.05)
        to(S1) delRepW(0.05) delRepE(0.05)
        to(S2) delRepW(0.05) delRepE(0.05)
        to(S3) delRepW(0.05) delRepE(0.05)
        to(S4) delRepW(0.05) delRepE(0.05)
        to(S5) delRepW(0.05) delRepE(0.05)
        to(S6) delRepW(0.05) delRepE(0.05)
        to(S7) delRepW(0.05) delRepE(0.05)
        to(S8) delRepW(0.05) delRepE(0.05)
        to(S9) delRepW(0.05) delRepE(0.05) moveN
        to(S10) delRepW(0.05) delRepE(0.05)
        to(S11) delRepW(0.05) delRepE(0.05) moveS
        to(S12) delRepW(0.05) delRepE(0.05)
        to(S13) delRepW(0.05) delRepE(0.05)
        to(S14) delRepW(0.05) delRepE(0.05)
        to(S15) delRepW(0.05) delRepE(0.05)
        to(S16) delRepW(0.05) delRepE(0.05)
        to(S17) delRepW(0.05) delRepE(0.05)
        to(S18) delRepW(0.05) delRepE(0.05)
        to(S19) delRepW(0.05) delRepE(0.05)
    stt S11
        to(S0) delRepW(0.05) delRepE(0.05)
        to(S1) delRepW(0.05) delRepE(0.05)
```

```
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05) moveN
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05) moveS
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S12
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05) moveN
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05) moveS
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S13
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05) moveN
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05) moveS
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
```

```
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S14
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05) moveN
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05) moveS
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S15
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05) moveN
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05) moveS
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S16
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
```

```
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05) moveN
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05) moveS
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05)
stt S17
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05) moveN
    to(S17) delRepW(0.05) delRepE(0.05)
    to(S18) delRepW(0.05) delRepE(0.05) moveS
    to(S19) delRepW(0.05) delRepE(0.05)
stt S18
    to(S0) delRepW(0.05) delRepE(0.05)
    to(S1) delRepW(0.05) delRepE(0.05)
    to(S2) delRepW(0.05) delRepE(0.05)
    to(S3) delRepW(0.05) delRepE(0.05)
    to(S4) delRepW(0.05) delRepE(0.05)
    to(S5) delRepW(0.05) delRepE(0.05)
    to(S6) delRepW(0.05) delRepE(0.05)
    to(S7) delRepW(0.05) delRepE(0.05)
    to(S8) delRepW(0.05) delRepE(0.05)
    to(S9) delRepW(0.05) delRepE(0.05)
    to(S10) delRepW(0.05) delRepE(0.05)
    to(S11) delRepW(0.05) delRepE(0.05)
    to(S12) delRepW(0.05) delRepE(0.05)
    to(S13) delRepW(0.05) delRepE(0.05)
    to(S14) delRepW(0.05) delRepE(0.05)
    to(S15) delRepW(0.05) delRepE(0.05)
    to(S16) delRepW(0.05) delRepE(0.05)
    to(S17) delRepW(0.05) delRepE(0.05) moveN
    to(S18) delRepW(0.05) delRepE(0.05)
    to(S19) delRepW(0.05) delRepE(0.05) moveS
stt S19
    to(S0) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S1) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S2) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S3) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S4) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S5) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S6) delRepW(0.05) delRepE(0.05) delRepS(0.05)
    to(S7) delRepW(0.05) delRepE(0.05) delRepS(0.05)
```

```
            to(S8) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S9) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S10) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S11) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S12) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S13) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S14) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S15) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S16) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S17) delRepW(0.05) delRepE(0.05) delRepS(0.05)
            to(S18) delRepW(0.05) delRepE(0.05) delRepS(0.05) moveN
            to(S19) delRepW(0.05) delRepE(0.05) delRepS(0.05)


results
    prob_S_00_00  = ((st LocX == 00) && (st LocY == 00)) ;
    prob_S_00_01  = ((st LocX == 00) && (st LocY == 01)) ;
    prob_S_00_02  = ((st LocX == 00) && (st LocY == 02)) ;
    prob_S_00_03  = ((st LocX == 00) && (st LocY == 03)) ;
    ...
    prob_S_19_18  = ((st LocX == 19) && (st LocY == 18)) ;
    prob_S_19_19  = ((st LocX == 19) && (st LocY == 19)) ;
```