



UNIVERSIDADE FEDERAL DE PERNAMBUCO - UFPE
CENTRO DE INFORMÁTICA - CIN
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**DESENVOLVIMENTO DE BENCHMARKS PARA SISTEMAS
MULTIAGENTE: O CASO DA PATRULHA ORIENTADA A
EVENTOS**

CANDIDATO
MARCELO JOSÉ SIQUEIRA COUTINHO DE ALMEIDA

TESE DE DOUTORADO
RECIFE, PE
FEVEREIRO DE 2013

Marcelo José Siqueira Coutinho de Almeida

**DESENVOLVIMENTO DE BENCHMARKS PARA SISTEMAS
MULTIAGENTE: O CASO DA PATRULHA ORIENTADA A
EVENTOS**

Tese apresentada à Coordenação do Programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, em cumprimento às exigências do Programa de Doutorado em Ciência da Computação.

Orientadores:
Patricia Tedesco
Geber Ramalho

Recife - Pernambuco – Brasil

Resumo

Sistemas Multiagente (SMA) tem se apresentado como uma abordagem eficiente no estudo de comportamentos inteligentes, bem como um importante paradigma de desenvolvimento de software distribuído e complexo. No entanto, a validação de seus produtos quase sempre é feita por meio de testes isolados, o que compromete o valor de seus resultados do ponto de vista científico. Uma solução para essa limitação é utilizar benchmarks, os quais são meios sistemáticos e rigorosos para estudar, planejar e incrementar técnicas e teorias por meio da comparação dos trabalhos desenvolvidos. No contexto da pesquisa científica, eles funcionam como instrumentos de validação, possibilitando medir o progresso de áreas onde os critérios de sucesso não são formais. Com o objetivo de demonstrar sua viabilidade na pesquisa em SMA, desenvolvemos um benchmark denominado PMAOE (*Event Oriented MultiAgent Patrolling*) baseado no problema da patrulha, o qual vem despertando atenção crescente da comunidade de SMA e Robótica. Uma vez que benchmarks ainda são desenvolvidos informalmente, desenvolvemos também um processo a fim de auxiliar pesquisadores a conduzir seu desenvolvimento de maneira sistemática.

Palavras-Chave: Avaliação de desempenho, Benchmarks, Sistemas Multiagentes, Patrulha Multiagente, Eventos.

Abstract

MultiAgent Systems (MAS) has emerged as an efficient approach in the study of intelligent behavior, as well as an important paradigm for distributed software development and complex. However, the validation of their products is almost always done by means of testing isolates, which undermines the value of its results from the viewpoint of scientific rigor. A solution to this limitation is to use benchmarks, which are systematic and rigorous ways to study, plan and improve techniques and theories by comparing the work done. In the context of scientific research, they serve as validation tools, allowing measure progress in areas where the success criteria are not formal. In order to demonstrate its viability in SMA research, we developed a benchmark called PMAOE (Event Oriented Patrolling multiagent) based on the patrolling problem, which is attracting increasing attention from the community of SMA and Robotics. Since that benchmarks are still developed in a non-systematic manner, we have also developed a set of methodological guidelines to assist researchers to conduct orderly and careful development.

Keywords: Performance Evaluation, Benchmarks, Multiagent Systems, Patrolling, Events.

“Toda a nossa ciência, comparada com a realidade, é primitiva e infantil – e, no entanto, é a coisa mais preciosa que temos.”

— Albert Einstein

Sumário

1. INTRODUÇÃO	1
2. SISTEMAS MULTIAGENTE E BENCHMARKS.....	6
2.1 Visão Geral de Benchmarks	6
2.2 Benchmarks em SMA	8
2.2.1 Benchmarks como Instrumento de Validação Científica.....	9
2.2.2 Elementos Básicos de Benchmarks em SMA	10
2.2.3 Valor Científico de Benchmarks	13
2.3 Tipos de Experimentos usando Benchmarks	15
2.4 Exemplos de Benchmarks em SMA.....	17
2.4.1 Robocup Soccer	17
2.4.2 RoboCup Rescue	18
2.4.3 TAC	18
2.5 Comentários Sobre o Capítulo	20
3. UM PROCESSO PARA DESENVOLVIMENTO DE BENCHMARKS.....	21
3.1 Justificativa.....	21
3.2 PROBENCH	24
3.2.1 Desenvolvimento	24
3.2.2 Visão Geral	28
3.2.3 Papéis Chave	28
3.3 Identificação da Classe de Problema.....	29
3.3.1 Descrição	29
3.3.2 Artefatos de Entrada	30
3.3.3 Atores	30
3.3.4 Atividades.....	31
3.3.5 Técnica Sugerida para Execução da Fase	33
3.3.6 Produto	34
3.4 Refinamento da Classe de Problema.....	34
3.4.1 Descrição	34
3.4.2 Entrada	34
3.4.3 Papéis Chave	34
3.4.4 Atividades.....	35
3.4.5 Técnica Sugerida	37
3.4.6 Artefato Produzido.....	37

3.5 Construção dos Datasets.....	37
3.5.1 Descrição	37
3.5.2 Artefatos de Entrada.....	37
3.5.4 Papéis.....	38
3.5.5 Atividades.....	38
3.5.5 Técnica Recomendada	41
3.5.6 Artefato Produzido.....	41
3.6 Definição das Métricas	41
3.6.1 Descrição	41
3.6.2 Artefatos de Entrada.....	42
3.6.3 Atores.....	42
3.6.4 Atividades.....	42
3.6.5 Técnicas Sugeridas	44
3.6.6 Produto	44
3.7 Escolha ou Desenvolvimento de um Simulador	45
3.7.1 Descrição	45
3.7.2 Artefatos de Entrada.....	45
3.7.3 Papéis.....	45
3.7.4 Atividades.....	46
3.7.5 Técnica Potencial	47
3.7.6 Produto	47
3.8 Conclusões sobre o Capítulo	47
4. IDENTIFICAÇÃO DA CLASSE DE PROBLEMA	49
4.1 Identificação dos Critérios de Seleção da Classe de Problema.....	49
4.2 Pesquisa por Classes de Problemas	52
4.3 Descrição da Classe de Problema.....	54
4.3.1 Agente Patrulhador.....	54
4.3.2 Região	55
4.3.3 Métricas	57
4.4 Análise dos Critérios de Escolha.....	58
4.5 Conclusões sobre o Capítulo	60
5. DETALHAMENTO DA CLASSE DE PROBLEMA	61
5.1 Identificação de Limitações	61
5.1.1 Falta de Entendimento Claro e Consensual de	62
5.1.2 Restrições na Comparabilidade.....	64
5.1.3 Cenários Super-simplificados.....	65
5.2 Aperfeiçoamento da Classe de Problema.....	66
5.2.1 Uma Nova Definição para Patrulha Multiagente	66

5.2.2 <i>Nossa Proposta: Patrulha Multiagente Orientada a Eventos</i>	67
5.2.3 <i>Eventos de Acordo com a Metafísica</i>	67
5.2.4 <i>Exemplo Ilustrativo</i>	70
5.2.5 <i>Discussão</i>	73
5.2.6 <i>Definição de Evento no Contexto da Patrulha</i>	74
5.3 <i>Modelagem da Classe de Problema</i>	78
5.3.1 <i>Justificativa</i>	78
5.3.2 <i>Classificação das Patrulhas Existentes</i>	80
5.3.3 <i>Taxonomia da Classe de Problema</i>	82
5.3.4 <i>Exemplos de Eventos</i>	88
5.3.5 <i>Taxonomia de Eventos</i>	90
5.3.6 <i>Comportamento dos Eventos</i>	97
5.4 <i>Conclusões sobre o Capítulo</i>	98
6. CONSTRUÇÃO DOS CENÁRIOS	100
6.1 <i>Descrição dos Cenários de Interesse</i>	100
6.1.1 <i>Grafo</i>	100
6.1.2 <i>Quanto à Sociedade</i>	102
6.1.3 <i>Quanto aos Eventos</i>	103
6.1.4 <i>Quanto aos Agentes</i>	104
6.1.5 <i>Quanto à Dinamicidade do Mundo</i>	104
6.2 <i>Identificação dos Atributos Primários e Secundários</i>	105
6.3 <i>Identificação de Restrições</i>	106
6.6 <i>Pesquisa Descrição de Problema</i>	107
6.5 <i>Pesquisa Literatura</i>	107
6.7 <i>Criação de Valores</i>	108
6.8 <i>Instanciação de Cenários</i>	108
6.10 <i>Conclusões sobre o Capítulo</i>	110
7. DEFINIÇÃO DAS MÉTRICAS	111
7.1 <i>Identificação dos Serviços dos Agentes</i>	111
7.2 <i>Identificação dos Requisitos de Serviço</i>	112
7.3 <i>Identificação das Métricas de acordo com Eficiência</i>	113
7.3.1 <i>Sucesso Baseado em Quantidade</i>	113
7.3.2 <i>Métricas Baseadas em Eficiência</i>	114
7.3 <i>Identificação das Métricas de acordo com a Responsividade</i>	116
7.5.1 <i>Atraso de Identificação</i>	117
7.5.2 <i>Métricas Básicas</i>	118
7.5.3 <i>Sucesso Baseado em Atraso</i>	120
7.3 <i>Identificação das Métricas de acordo com Impacto</i>	123

7.4 Análise Comparativa entre PMAOE e Benchmarks Existentes.....	126
7.6 Conclusões sobre o Capítulo	130
8. ESTUDO EXPERIMENTAL	132
8.1 SIMPATROL	132
8.2 Agentes Estudados	133
8.3 Metodologia	134
8.4 Experimento 1: Afogamento	134
8.3.1 <i>Dataset</i>	134
8.3.2 <i>Resultados e Discussão</i>	137
8.4 Experimentos 2: Colisão de Automóveis	138
8.4.1 <i>Dataset</i>	138
8.4.2 <i>Resultados e Discussão</i>	140
8.5 Experimento 3: Explosão em Caldeira.....	141
8.5.1 <i>Dataset</i>	141
8.5.2 <i>Resultados e Discussão</i>	142
Conclusões sobre o Capítulo	143
9. CONCLUSÕES E TRABALHOS FUTUROS.....	144
9.1 Resumo do Trabalho	144
9.2 Principais Contribuições	146
9.3 Trabalhos Futuros.....	147
REFERÊNCIAS	149

Lista de Figuras

Figura 1.1: Organização do documento.....	5
Figura 2.1: Relação entre os componentes de um benchmark em SMA.....	10
Figura 2.2: Relação entre soluções e cenários na realização de experimentos.....	16
Figura 3.1: Fases de PROBENCH.....	24
Figura 3.2: Seleção da classe de problema.	31
Figura 3.3: Refinamento da classe de problema.....	35
Figura 3.4: Construção de cenários.	40
Figura 3.5: Instanciação de cenários.	41
Figura 3.6: Definição das métricas.	44
Figura 4.1: Quantidade de trabalhos em PMA no período de 2002 a 2012.	54
Figura 4.2: Processo de Esqueletonização.....	56
Figura 5.1: Estratégia de visita na patrulha cega.	71
Figura 5.2: Estratégia de visita na patrulha com conhecimento.....	72
Figura 5.3: estratégia de visita baseado em fator de impacto e probabilidade de ocorrência.	73
Figura 5.4: Relação temporal entre eventos.	76
Figura 5.5: Relação entre tipos de patrulhas em	81
Figura 5.6: Visão de alto nível da Patrulha Multiagente.	82
Figura 5.7: Caracterização de um grafo no problema	83
Figura 5.8: Caracterização de uma sociedade no problema da patrulha.	85
Figura 5.9: Representação de um agente patrulhador.....	87
Figura 5.10: Caracterização das métricas em	88
Figura 5.11: Visão de alto nível da taxonomia de eventos.....	92
Figura 5.12: Representação da origem de um evento.....	93
Figura 5.13: Representação da ocorrência de um evento.	94
Figura 5.14: Representação de pré-condições de um evento.....	94
Figura 5.15: Representação de impacto de um evento.	95
Figura 5.16: Representação da reação a um evento.....	96
Figura 5.17: Representação da consequência a um evento.	97
Figura 7.1: Representação de um evento do ponto de vista de sua duração e do instante de visita do agente ao vértice.....	114
Figura 7.2: Atraso de identificação.....	117
Figura 7.3: Função sucesso para eventos evitáveis.	121
Figura 7.4: Função sucesso para eventos manipuláveis.	122

Figura 7.5: Função sucesso para eventos tratáveis.	122
Figura 7.6: Representação do fator de impacto.	124

Lista de Quadros

Quadro 3.1: Relação entre trabalhos sobre benchmarks em SC.....	26
Quadro 5.1: Classificação dos eventos de acordo com o momento adequado de visita.	98
Quadro 6.1: Restrições dos atributos de PMAOE.	107
Quadro 6.2: Valores de grafo.	108
Quadro 6.3: Valores de sociedade.	109
Quadro 6.4: Valores de agente.	110
Quadro 7.1: Análise comparativa dos benchmarks em SMA.	130
Quadro 8.1: Dataset utilizado no experimento 1.	135
Quadro 8.2: Porcentagem média dos eventos detectados.....	137
Quadro 8.3: Atraso médio dos eventos detectados.....	138
Quadro 8.4: Dataset utilizado no experimento 1.	138
Quadro 8.5: Porcentagem média dos eventos detectados.....	140
Quadro 8.6: Atraso médio dos eventos detectados.....	140
Quadro 8.7: Dataset para o experimento 3.	141

Lista de Equações

Equação 7.1: Quantidade Total de Eventos Identificados.....	115
Equação 7.2: Quantidade Total de Eventos Evitados.....	115
Equação 7.3: Quantidade Total de Eventos Manipulados.....	116
Equação 7.4: Quantidade Total de Eventos Tratados.....	116
Equação 7.5: Quantidade Total Relativa de Eventos Identificados.....	116
Equação 7.6: Função atraso de identificação.....	117
Equação 7.7: Atraso mínimo experimentado por um agente.....	118
Equação 7.8: Atraso máximo experimentado por um agente.....	118
Equação 7.9: Atraso médio experimentado por um agente.....	119
Equação 7.10: Atraso mínimo experimentado pela sociedade S.....	119
Equação 7.11: Atraso máximo experimentado por uma sociedade S.....	119

Lista de Acrônimos

AAAI:	Association for Advancement of Artificial Intelligence
B2B:	Bussiness-to-Bussiness
B2C:	Bussiness to Consumer
CAS:	Continuous Area Sweeping
CC:	Conscient Cognitive (Cognitivo Consciente)
CR:	Conscient Reactive (Reativo Consciente)
PMAOE:	Event Based Multiagent Patrolling
HPCC:	Heuristic Path Conscient Cognitive
IA:	Inteligência Artificial
IAD:	Inteligência Artificial Distribuída
IBM:	International Bussiness Machines
ICGA:	International Computer Games Association.
IDH:	Índice de Desenvolvimento Humano.
MIT:	Massachussets Institute of Technology
NIST:	National Institute of Standards and Technology
PIB:	Produto Interno Bruto
PLN:	Processamento de Linguagem Natural
:	Patrulha Multiagente.
MATP:	Patrulha Multiagente Temporal.
PMAOE:	Patrulha Multiagente Orientada a Eventos
PMI:	Project Management Institute
PMR:	Patrulha Multirobô
PNT:	Propriedade Não-Tangencial
PT:	Propriedade Tangencial
QI:	Quociente de Inteligência
RCS:	Robocup Soccer
RCR:	Robocup Rescue
SED:	Simulação de Eventos Discretos
SMA:	Sistemas Multiagente.
TAC:	Trading Agent Competition
TAC-AA:	Trading Agent Competition Ad Auction.
TAC-SGM:	Trading Agent Competition Manufacturer.
UML:	Unified Modeling Language

Capítulo 1

Introdução

Ao longo das últimas seis décadas, o campo da Inteligência Artificial (IA) tem contribuído positivamente para o desenvolvimento de tecnologias importantes, tais como mecanismos de busca, sistemas especialistas, sistemas de navegação, sistemas de reconhecimento de imagens, e jogos eletrônicos (Russel e Norvig, 2004). Dentre suas **subáreas**, Sistemas MultiAgente (SMA) tem se apresentado como uma abordagem eficiente no estudo de comportamentos inteligentes, bem como um importante paradigma de desenvolvimento de software distribuído e complexo. Apesar disto, a dificuldade em medir o progresso da área representa um obstáculo para sua consolidação como disciplina científica. Essa dificuldade está relacionada principalmente com o fato de que, por se tratar de uma ciência do artificial, torna-se complexo definir critérios de sucesso formalmente. Portanto, faz-se necessário desenvolver meios adequados para avaliar SMA.

Uma alternativa que tem se mostrado bastante eficiente é a adoção de benchmarks (Hanks et al., 1993; Hayes e Ford, 1995; French, 2000; Sim, 2006; Drogoul et al., 2007;). Benchmarks proveem meios sistemáticos e rigorosos para se estudar, planejar, analisar, desenvolver, compreender, comparar, selecionar e melhorar teorias e técnicas. Neste sentido, um benchmark consiste em um cenário padronizado contendo um problema que precisa ser resolvido usando algum tipo de comportamento inteligente.

A importância de benchmarks em IA está relacionada com a própria natureza da área, a qual se fundamenta na observação de comportamentos inteligentes na natureza e na tentativa de repeti-los artificialmente (Newell e Simon, 1984). Os modelos desenvolvidos a partir dessa observação são projetados, implementados e testados em dispositivos artificiais (Simon, 1982). Portanto, em IA, o processo de experimentação deve ser realizado através de simulação da solução proposta ao problema em um cenário simplificado, para que pesquisadores possam avaliar o comportamento de suas teorias e/ou técnicas (Hanks et al., 1993).

Outra razão pela qual benchmarks são importantes é que eles servem como

forma de testar uma solução para um problema complexo e que demanda esforços de longo prazo. Dessa forma, eles funcionam como um desafio e despertam a atenção da comunidade para a solução do problema e, portanto, tornam-se uma maneira de direcionar a agenda de pesquisas (Kitano et al., 1996; Stone, 2003; Drougoul et al., 2007; Rohrer, 2009).

Mesmo dentro de uma área específica como IA, faz-se necessária a existência de diversos benchmarks, a fim de apoiar o avanço do estado da arte em cada subárea do conhecimento, tais como SMA, planejamento, aprendizagem de máquinas, raciocínio automático, casamento de ontologias etc.

Apesar de sua importância como paradigma de desenvolvimento de software e de seu potencial no estudo de comportamentos inteligentes e distribuídos, poucos benchmarks têm sido desenvolvidos nesse contexto, e mesmo os existentes não atendem de forma satisfatória às necessidades de pesquisa (Huppler, 1999; Bayer et al., 2004; Prechelt, 1994; Stone, 2003; Duchateau, 2006; Drogoul, 2007). Dentre esses podemos citar *RoboCup Soccer* (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009), *RoboCup Rescue* (Kitano et al., 1999; Takahashi, 2009) e *Trading Agent Competition* (Wellman e Wurman, 1999; Vetsikas e Selman, 2002; Cai et al., 2009; Niu et al. 2009).

Uma alternativa promissora é a Patrulha Multiagente (PMA), a qual consiste em um grupo de agentes percorrendo uma área restrita a fim de monitorar ou supervisionar todos seus locais (Abate, 1996). PMA tem sido usada tanto no estudo de técnicas voltadas a problemas de IA como de robótica, abrangendo domínios como aplicações militares, supervisão de patrimônio, resgate de pessoas perdidas e gerência de redes de computadores. Adicionalmente, ela tem sido usada no estudo de comportamentos inteligentes importantes, tais como planejamento, negociação, coordenação, comunicação, cooperação, aprendizado e tomada de decisão. Entretanto, existem três limitações que impedem sua consolidação como benchmark para o estudo de SMA.

A primeira limitação diz respeito à *falta de um entendimento claro e consensual do que é PMA como classe de problema*. Apesar de sua aplicabilidade e potencial como benchmark, a pesquisa atual é confusa e desorganizada. A adoção de uma definição informal em sua fundamentação teórica tem levado ao surgimento de variações da tarefa da patrulha, as quais são muito diferentes entre si em termos de objetivos dos agentes, métricas de desempenho, características e restrições do ambiente. Por exemplo, enquanto em alguns trabalhos os agentes buscam localizar um invasor, outros se concentram em minimizar os intervalos entre visitas a cada vértice.

A segunda limitação diz respeito à *comparabilidade*. Cada trabalho segue abordagens distintas, e não há qualquer critério de comparação. Por exemplo, enquanto alguns trabalhos lidam com topologias bastante simples, como por exemplo um corredor ou um círculo, outros lidam com topologias irregulares e com uma grande

quantidade de vértices. Isso decorre da falta de uma descrição padronizada das propriedades que caracterizam como classe de problema e suas diferentes variações.

A terceira limitação é a *super-simplificação dos problemas estudados*. Por exemplo, os grafos utilizados em vários trabalhos são bastante simples quando comparados às regiões que devem ser patrulhadas no mundo real. Em alguns casos é utilizado um grafo circular (Agmon, 2010) ou uma linha (Chevaleyre, 2004). Também a ideia de realizar uma patrulha sem dispor de nenhuma informação sobre o ambiente não condiz com aquelas comumente utilizadas na prática.

Dessa forma, o objetivo principal desta tese é o desenvolvimento da PMAOE (Patrulha MultiAgente Orientada a Eventos), um benchmark voltado para de SMA por meio do problema da patrulha. Diferente das abordagens existentes, PMAOE considera que os agentes possuem conhecimento explícito do mundo, sendo capazes de escolher os locais que serão visitados de acordo com o estado corrente da região e as características de ocorrência dos eventos.

É importante ressaltar que o desenvolvimento de benchmarks é uma atividade complexa. Ao desenvolvê-los, pesquisadores se deparam com questões tais como: qual classe de problemas deve ser estudada? Quais conceitos são pertinentes à classe de problema escolhida? Quais cenários devem ser utilizados em cada experimento? Quais métricas são mais apropriadas para avaliar o desempenho de cada solução?

Apesar dessa constatação, não conseguimos identificar na literatura processos, métodos ou diretrizes detalhadas que auxiliem pesquisadores na criação de benchmarks. Por um lado, isso faz com que sejam adotadas práticas totalmente informais e individualizadas, comprometendo a validade dos resultados obtidos nos experimentos, e por outro, o compartilhamento de esforços e o reuso de conhecimento são dificultados (Zöller et al., 2006).

Dessa forma, a partir da experiência em PMAOE, desenvolvemos um processo que visa auxiliar pesquisadores na criação de benchmarks. Isso foi realizado a partir da análise criteriosa sobre quais artefatos são necessários no estudo de um sistema computacional e como cada um deles deve ser criado.

É importante ressaltar que, do ponto de vista epistemológico, nossa proposta se insere no campo da metodologia da pesquisa científica e do reducionismo metodológico e ontológico (Nagel, 1998; Feyerabend; 1998; Nickles, 1998; Kitcher, 1998). Isso significa dizer que este trabalho objetiva, além do desenvolvimento de abordagens que auxiliem na condução de experimentos em de SMA, simplificar seu estudo por meio da unificação dos problemas existentes em PMA.

As principais contribuições desta tese são a organização conceitual do problema da patrulha; o desenvolvimento de um benchmark para o estudo de SMA; e um processo de criação de benchmarks.

O restante deste documento está organizado conforme apresentado na figura 1.1. Nela temos o seguinte: o *problema* é tratado no capítulo 2 (“Sistemas MultiAgente e Benchmarks”) e no capítulo 4 (“Identificação da Classe de Problema”), a *hipótese* é tratada no capítulo 3 (“Diretrizes Metodológicas para o Desenvolvimento de Benchmarks”). Os capítulos 4 (“Identificação da Classe de Problema”), capítulo 5 (“Refinamento da Classe de Problema”), capítulo 6 (“Construção dos Cenários”) e capítulo 7 (“Definição das Métricas”) demonstram a aplicabilidade do processo proposto durante a criação de um benchmark para SMA. No capítulo 8 (“Estudo Experimental”) são descritos os experimentos desenvolvidos e seus resultados. Por fim, no capítulo 9 (“Conclusão e Trabalhos Futuros”) são apresentados um resumo da pesquisa, as contribuições obtidas e uma discussão acerca das atividades futuras.

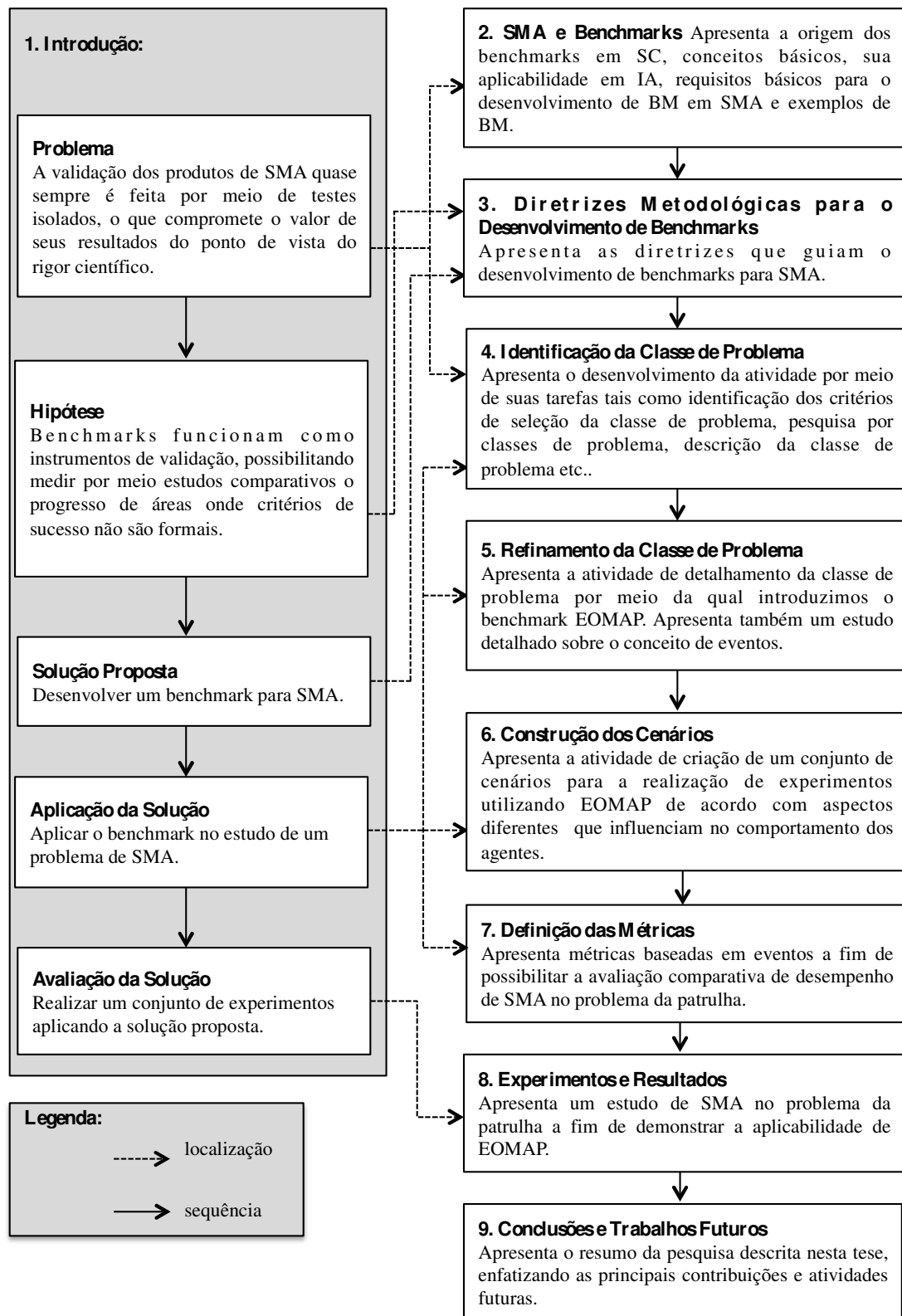


Figura 1.1: Organização do documento.

Capítulo 2

Sistemas Multiagente e Benchmarks

SMA tem se apresentado como uma abordagem eficiente no estudo de comportamentos inteligentes, bem como um importante paradigma de desenvolvimento de software distribuído e complexo. No entanto, a validação de seus produtos quase sempre é feita por meio de testes isolados, o que compromete o valor de seus resultados do ponto de vista do rigor científico. Uma solução para essa limitação é testar os trabalhos realizados de forma comparativa. Uma alternativa viável é a adoção de benchmarks, os quais permitem observar o desempenho de um conjunto de soluções usando o mesmo problema, configurações de testes, e observado as mesmas métricas. Portanto, benchmarks fornecem o aparato científico necessário para consolidar de forma rigorosa os produtos desenvolvidos em SMA.

Este capítulo tem três objetivos principais: O primeiro é contextualizar a pesquisa realizada nessa tese através de uma revisão sobre benchmarks, ressaltando as diferenças entre essa prática em administração e em SMA; O segundo é apresentar conceitos básicos de benchmarks, a fim de facilitar o entendimento dos capítulos seguintes; O terceiro objetivo é elencar requisitos fundamentais que devem ser atendidos por benchmarks em SMA.

2.1 Visão Geral de Benchmarks

Do ponto de vista da Tecnologia da Informação (TI), o termo benchmark tem sua origem no contexto da administração de empresas, mais especificamente na área de Planejamento Estratégico, na década de 1960 (Boxwell, 1996). De acordo com Spendolini (1993), benchmarks são processos contínuos e sistemáticos de avaliação de produtos, serviços e processos de trabalho com a finalidade de melhoria organizacional. Ainda de acordo com o mesmo, um benchmark pode ser visto por meio de dois principais aspectos: avaliação de desempenho e aprendizado.

Do ponto de vista da *avaliação de desempenho*, benchmarks permitem observar, medir e comparar valores que representam como o sistema se comporta durante a realização de uma determinada tarefa. Indivíduos, instituições, equipes esportivas,

governos ou grupos de pesquisa são comparados com base em algum critério, tais como notas de exames escolares, Quociente de Inteligência (QI), quantidades de partidas vencidas em um campeonato, Índice de Desenvolvimento Humano (IDH), Produto Interno Bruto (PIB) etc. No caso específico das organizações, medições são utilizadas com o objetivo de auxiliar o processo de tomada de decisão, possibilitando que os responsáveis pela sua administração identifiquem as alternativas mais promissoras na busca pela melhoria da qualidade.

Do ponto de vista do *aprendizado*, benchmarks permitem que uma empresa aprenda com as melhores práticas realizadas por empresas consideradas como referência em um determinado segmento. Ainda de acordo com Spendolini (1993), benchmarks, por si só, não dão respostas sobre como processos podem ser aperfeiçoados. Benchmarks fornecem resultados sobre como algum processo é conduzido na empresa e, a partir daí, os responsáveis devem comparar seus resultados com os das empresas de referência e decidir o que deve ser feito: manter ou modificar os procedimentos da organização.

Ainda na década de 1960, o sucesso dos benchmarks rapidamente chamou a atenção de outras áreas, e sua prática começou a ser disseminada. Mais especificamente, no que diz respeito à Computação, surgiram benchmarks para redes de computadores, concepção de circuitos integrados, arquiteturas de computadores e desenvolvimento de software (John, 2006).

No entanto, é preciso ressaltar que existem diferenças substanciais no entendimento e na forma como benchmarks são aplicados em cada uma das áreas. Em Administração, a aplicação de benchmarks se concentra em descobrir quais procedimentos devem ser adotados a fim de alcançar o nível de excelência de empresas consideradas como referência em uma determinada prática. Em Computação, o objetivo é medir o desempenho de uma solução tomando como base uma ou mais soluções concorrentes. Adicionalmente, nos benchmarks em administração são utilizados métodos analíticos (observação, entrevistas, formulários etc.) como forma de coletar informações sobre como uma empresa realiza suas atividades, e identificar o que deve ser realizado a fim de torná-la mais competitiva. Na computação, geralmente são utilizados métodos quantitativos através de ferramentas de avaliação de desempenho, a fim de observar como o sistema estudado se comporta em determinadas situações. Os resultados produzidos são então comparados com os das outras soluções.

Em Zöller et al. (2006) é feita uma comparação entre a atividade de Gestão de Processos de Negócios (*Business Process Management* - BPM) e a de Desempenho Computacional (DC) (Quadro 2.1), na qual podemos inserir também SMA.

São utilizados quatro aspectos: objeto, critério, método e referência. No primeiro aspecto (*objeto*), benchmarks em BPM focam em processos de negócio, enquanto DC foca em sistemas computacionais, cujos itens avaliados vão desde a capacidade de

realizar operações em ponto flutuante até a capacidade de um servidor de grande porte em processar transações (John, 2006). No segundo aspecto (*critério*), em BPM são utilizados tempo, custo e qualidade, enquanto em DC os critérios estão relacionados com a capacidade do sistema em analisar operações em um intervalo de tempo especificado. No terceiro aspecto (*método*), é empregada a análise dos processos da organização estudada enquanto em desempenho computacional são executadas rotinas que testam o sistema em estudo utilizando informações que representam cenários que imitam o mundo real em um determinado contexto. Por último, no quarto aspecto (*referência*), em BPM, é adotada a referência qualitativa enquanto que em DC é a quantitativa.

Quadro 2.1: Comparação entre benchmarks em BPM e DC (Zöller et al., 2006)

	BPM	DC
Objeto	Processos de negócio	Sistemas computacionais
Critério	Tempo, custo, qualidade	Tempos de vazão
Método	Análise do processo	Execução de rotinas de benchmark
Referência	Qualitativa	Quantitativa

Dentre os tipos de sistemas computacionais (SC) que adotam benchmarks como método de avaliação, destacamos a IA. A partir da próxima seção iremos nos concentrar especificamente em analisar a importância de benchmarks no contexto específicos de SMA, a fim de não nos desviarmos dos objetivos desta tese. No entanto, deve-se ressaltar que muitas das conclusões e contribuições apresentadas podem ser generalizadas para outras áreas de SC.

2.2 Benchmarks em SMA

A preocupação em validar soluções e medir o progresso é extremamente importante em qualquer área da ciência. Duas maneiras de fazê-lo são através da comparação de soluções desenvolvidas e da replicação de experimentos. No caso particular de SMA, devido à sua natureza multidisciplinar e complexidade dos sistemas estudados, realizar essas atividades se torna frequentemente difícil.

Conforme relatado na seção 2.1, benchmarks vêm sendo usados com sucesso em SC. No caso específico da IA, várias sub-áreas vêm adotando essa prática, tais como Redes Neurais (Prechelt, 1994), Mineiração de Dados (Narayanan et al., 2006), Aprendizagem Automática (Zheng, 1993), Web Semântica (Guo et al., 2006), Raciocinadores (Gardiner et al., 2006; Ferrara et al., 2011), Planejamento (Howe e Dahlman, 2002), e Problemas de Satisfação de Restrições (Ramani et al., 2006). No

entanto, no contexto de SMA é preciso compreender melhor o papel dessa abordagem, a fim de expandir seu uso e torná-lo uma prática comum dentro das pesquisas da área.

2.2.1 Benchmarks como Instrumento de Validação Científica

IA é uma ciência que lida especificamente com o artificial (Simon, 1996). Diferente de ciências como Física e Química, em que teorias e modelos desenvolvidos se aplicam a um domínio do mundo natural, IA não dispõe de meios naturais para testar suas teorias. Por outro lado, apesar das semelhanças com a Psicologia, ao desenvolver teorias e modelos que descrevem o comportamento humano (ou de outros tipos de seres vivos), IA se propõe a implementá-los em sistemas computacionais. Sendo assim, a fim de ser considerada válida, uma teoria deve ser implementada e demonstrada através da *ação inteligente* de um sistema de software no mundo (Bittencourt, 1996).

O uso de softwares específicos por meio dos quais o comportamento dos sistemas é simulado tem sido muito comum. Simulação é o processo de projetar um modelo de um sistema real e conduzir experimentos, a fim de entender seu comportamento de acordo com os limites impostos por critérios de operação (Shannon, 1975). Através deste recurso é possível estudar teorias e modelos, observando os efeitos de variações nas hipóteses e, conseqüentemente, comprová-las ou refutá-las (Hanks et al., 1993).

No entanto, a realização de testes de forma isolada não oferece o rigor científico necessário, sobretudo no que diz respeito à replicabilidade e comparabilidade. É fundamental que tanto os parâmetros como os artefatos de teste envolvidos sejam compartilhados pela comunidade científica. Uma solução para esse problema é a adoção de benchmarks.

De acordo com Sim (2003), um benchmark operacionaliza um paradigma científico. Isso significa dizer que um benchmark é capaz de incorporar, organizar e tornar manipulável o conhecimento que rege um determinado domínio do conhecimento, fazendo-o avançar por meio de análises comparativas. Essa forma de realizar ciência vai ao encontro da compreensão apresentada por Kunn (2010) acerca do processo evolutivo de realizar ciência. Segundo o mesmo, a ciência evolui por meio de revoluções, através das quais o conhecimento comum é revisado ou adaptado. Cada processo de revisão do paradigma dominante por meio de novas descobertas consiste em uma revolução.

O relacionamento entre benchmarks e o paradigma científico dominante para uma disciplina forma a base para a existência de mecanismos que trazem os efeitos positivos dos benchmarks: em primeiro lugar, benchmarks promovem um modelo aberto de ciência que convida a comunidade envolvida ao escrutínio e à colaboração;

Em segundo lugar, benchmarks incentivam a fertilização cruzada de idéias entre os pesquisadores e os diferentes papéis que eles podem assumir (Sim, 2003).

2.2.2 Elementos Básicos de Benchmarks em SMA

Um benchmark em SMA consiste nos seguintes elementos: *classe de problema*, *simulador*, *cenários*, e *métricas* (Figura 2.1). Esses elementos foram observados nos benchmarks que analisamos na literatura, conforme será apresentado na seção 3.2. A fim de realizar um experimento, o pesquisador deve projetar um modelo que descreva alguma hipótese acerca da classe de problema, implementá-lo usando alguma linguagem de programação e testá-lo através do simulador (Cohen et al., 1993). Cada solução será testada para cada variação do cenário, o qual representa uma instância do problema. O simulador irá coletar e gerar um conjunto de resultados. Os resultados gerados serão utilizados pelos pesquisadores a fim de comparar o desempenho em cada cenário e com as soluções concorrentes.

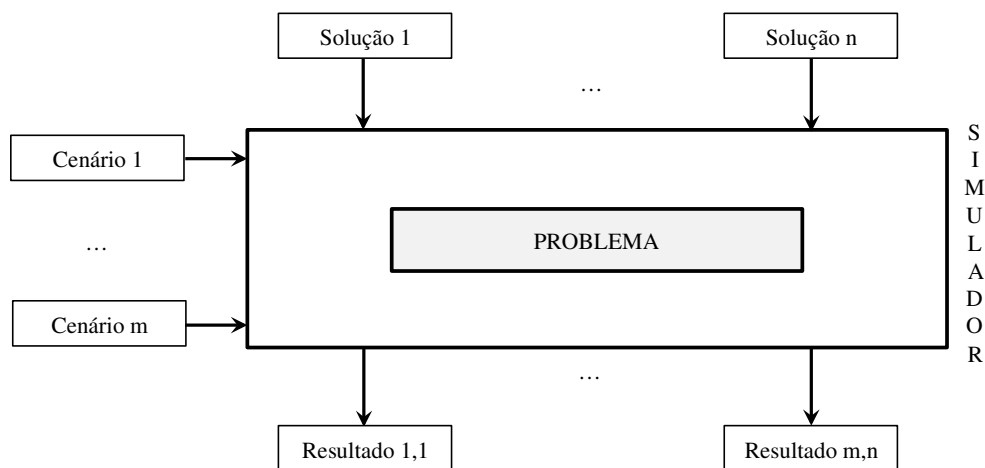


Figura 2.1: Relação entre os componentes de um benchmark em SMA.

Classe de problema: descreve um tipo de problema que pode ser utilizado para a realização de pesquisas. Ela deve permitir a exploração de diversas técnicas e/ou estratégias, a fim de atender às necessidades dos pesquisadores daquela área. Uma dessas técnicas é verificar o desempenho da aplicação desenvolvida durante a realização da tarefa envolvida.

A classe de problema deve consistir em uma amostra representativa das tarefas que um conjunto de ferramentas ou técnicas deve ser capaz de resolver na prática. Uma vez que não é possível incluir o conjunto inteiro de tarefas do domínio de problema (e. g., SMA), uma seleção de tarefas funciona como um substituto (Sim, 2003). Como

exemplo de classes de problema podemos citar o *caixeiro viajante*, *torres de Hanói* e *ordenação de vetores* (John, 2006).

Simulador: é um software que fornece funções e recursos para realização de estudos usando modelos que imitam o funcionamento de sistemas reais. Dessa forma, simuladores permitem estudar a viabilidade de um sistema e analisar seu desempenho antes dele ser construído. Dentre as funções comumente disponibilizadas estão controle da fila de eventos, simulação de relógio, troca de mensagens, atualização e armazenamento das medições. Além disso, o simulador deve fornecer entidades de software que representem os elementos da classe de problema. Um simulador deve ser capaz de receber um conjunto de valores de entradas que representam a configuração do experimento, executar o sistema em estudo, calcular ou coletar os valores das métricas e registrá-los em algum formato que possa ser depois processado.

Cenário: representa uma condição particular do mundo, e é descrito por meio de um conjunto de parâmetros os quais descrevem restrições impostas às soluções sendo desenvolvidas. A fim de representar diferentes níveis de complexidade a que a solução deve ser submetida, os cenários devem ser variados. Por exemplo, no trabalho de Pronobis et al. (2010), variações nos parâmetros consistem em modificar o ambiente (corredor, sala), a iluminação (claro, escuro), o clima (sol, neblina, chuva, neve) etc.

A disponibilidade de cenários é fundamental para a realização de experimentos. Em alguns casos, pode-se fazer uso daqueles disponíveis na literatura. No entanto, isso nem sempre é possível, pois em muitos casos os autores não disponibilizam as informações de forma clara e completa. Nesse caso, cenários podem ser criados através da instanciamento dos atributos que descrevem a classe de problema de acordo com os objetivos estabelecidos (John, 2006; Zöllner et al., 2006).

Em alguns tipos de problemas, tais como em aprendizagem automática (Prechelt, 1994), cenários são classificados em duas ou mais partes: *treinamento*, *validação* e *teste*. A solução em questão, uma rede neural, utiliza os dados de treinamento para descobrir qual função, dentre um conjunto de funções pré-definidas, será utilizada para resolver a tarefa especificada. Em seguida, utilizando os dados de *validação*, verifica-se se a rede neural atende aos critérios de desempenho. Daí o sistema estará pronto para analisar qualquer base de dados de *teste*.

Conforme discutido por Howe e Dahlman (2002), não é prático, ou sequer desejável, executar em um mesmo experimento todos os cenários possíveis. Para isso é preciso escolher de maneira informada aqueles cenários que mais se adequam ao propósito da pesquisa. Ainda segundo os mesmos autores, alguns sistemas foram desenvolvidos para atender a problemas específicos. Dessa forma, esses sistemas poderão desempenhar de maneira imprevisível em outros cenários.

É preciso analisar a necessidade de equilibrar a relação entre generalidade ou especificidade na solução, e desempenho obtido. Isso significa dizer que, em alguns casos, pode ser aceitável que a solução tenha um melhor desempenho em alguns cenários e pior em outros. Em outros casos, pode ser interessante ter uma solução que atenda razoavelmente bem às expectativas de desempenho em todos os cenários.

Ainda de acordo com Howe e Dahlman (2002), outro ponto que requer atenção em relação à escolha dos cenários é que muitos deles devem ser descartados porque são simples demais, distantes das aplicações reais. De forma semelhante, determinadas combinações dos valores dos parâmetros não se traduzem em cenários que façam sentido no mundo real ou valham a pena serem investigados do ponto de vista teórico.

Finalmente, conforme discutido na seção 2.5, benchmarks têm que evoluir conforme o estado da arte da área progredida, e suas soluções sejam consolidadas e adotadas no mercado ou indústria. Portanto, a evolutibilidade de um benchmark em muito depende da atualização dos problemas sendo estudados e, conseqüentemente, da atualização dos cenários ou instâncias de problema.

Do ponto de vista prático, um cenário é representado por um ou mais *datasets*. Um *dataset* representa uma instância do cenário que é usado na realização de experimentos práticos.

Métrica: descreve um critério de desempenho associado ao comportamento do modelo em estudo através do qual é realizada uma avaliação quantitativa ou qualitativa de sua capacidade em realizar uma determinada tarefa (Howe e Dahlman, 2002).

Métricas devem ser escolhidas dependendo dos objetivos do benchmark (Solingen, 1999; Zöller et al., 2009). Seus objetivos devem estar relacionados com as capacidades, recursos ou funcionalidades de interesse do sistema sendo estudado. Métricas servem, portanto, para verificar como o sistema se comporta em determinados cenários do ponto de vista de seu desempenho. Por outro lado, de acordo com Wörner e Wörn (2006), o desempenho do simulador é diretamente afetado pelo número de métricas. Sendo assim, dependendo da quantidade e do tipo dos cálculos envolvidos na medição do desempenho, o simulador pode demandar mais recursos de processamento e memória.

Alguns exemplos de métricas de desempenho que encontramos na literatura são tempo de resposta, utilização de recursos, esforço, tempo de carga, frequência de conflitos para logística de manufatura (Zöller et al., 2006), e número de operações aritméticas usando inteiros ou em ponto flutuante por segundo, taxa de compressão de dados, número de transações realizadas por segundo, tempo para renderizar uma imagem e, concepção de hardware (John, 2006).

Métricas tem influência direta na *evolutibilidade* do benchmark. Elas atendem aos desafios que forem surgindo ao longo do tempo, o que demanda a remoção, modificação ou introdução de novas métricas.

2.2.3 Valor Científico de Benchmarks

De acordo com Cohen e colegas (1993), o valor científico de benchmarks e simuladores está no poder de ressaltar aspectos importantes do desempenho do sistema. O controle experimental que pode ser alcançado com simuladores pode auxiliar a explicar porque o sistema se comporta de uma forma ou de outra. Esse tipo de abordagem, apesar de necessária, leva a uma tensão entre duas alternativas: por um lado, sistemas simples são mais fáceis de controlar, porém reduzem o poder dedutivo do experimento, explicando menos fenômenos e comprometendo a generalização da teoria. Portanto, o pesquisador deve encontrar o equilíbrio entre controle do experimento e poder explanatório, e de generalização da teoria.

Após a realização de um benchmarking, o pesquisador deve ser capaz de responder a perguntas como:

- O resultado obtido é o esperado?
- O que o resultado significa do ponto de vista das alternativas escolhidas para a implementação dos agentes e da definição do ambiente?
- Como o resultado da solução desenvolvida se compara com os de outras existentes?
- Os resultados são suficientes para entender o comportamento da solução?
- É possível identificar a consequência de cada parte da solução nos resultados obtidos?
- É possível identificar como o ambiente influencia nos resultados obtidos?
- É possível confirmar ou refutar claramente a hipótese?
- É possível repetir o experimento em condições mais flexíveis?

Um benchmark deve abordar questões importantes e fornecer desafios que promovam o estado da arte da área. No entanto, poucos benchmarks têm sido desenvolvidos com esse intuito, havendo uma certa tendência na utilização de simuladores isoladamente. Dessa forma, podemos classificar os benchmarks em dois tipos: **simulação individual** e **benchmark padronizado**.

No primeiro caso, cada pesquisador ou grupo de pesquisadores se dedica a resolver um problema específico, definindo métricas e cenários de acordo com suas necessidades particulares. Em seguida, eles utilizam um Simulador de Propósito Geral (SPG) ou criam seus próprios simuladores para realizar os experimentos. Como

exemplos de SPG temos JASON (Bordini e Hübner, 2009), STARLOGO (Resnick, 1994), TURTLEKIT (Michel et al., 2005) e SESAM (Klügl, 2009). Podemos observar as seguintes limitações nessa abordagem:

- **Falta de compartilhamento de esforços:** uma vez que benchmarks individuais não dispõem de problemas compartilhados, todo esforço inicial para elaborar cenários, definir métricas, escolher metodologias e simulador geralmente começa do zero. Por outro lado, o uso de benchmarks padronizados promove a produtividade porque eles aceleram o processo de pesquisa, uma vez que os pesquisadores podem focar na busca da solução em si, e não na definição do problema;
- **Impossibilidade de comparação:** uma comparação só pode ser feita entre soluções diferentes se elas compartilham o mesmo problema e nas mesmas condições. Ou seja, cada solução deve ser desenvolvida levando em consideração os mesmos cenários, restrições, recursos e métricas. Adotar um benchmark individual dificulta a realização de pesquisas nas mesmas condições e, conseqüentemente, sua comparação;
- **Impossibilidade de replicação de estudos:** a replicação de experimentos e, conseqüentemente, validação das soluções apresentadas pelos pesquisadores depende essencialmente do compartilhamento dos cenários de experimentação, das ferramentas utilizadas, e da disponibilidade de dados das demais pesquisas. Sendo assim, o uso de benchmarks individuais inviabiliza a replicação de resultados.

Em relação aos benchmarks padronizados, temos que uma comunidade de pesquisadores reúne esforços, mesmo que informalmente, a fim de promover o estado da arte de uma área. Em outras palavras, nesse tipo de benchmark, a comunidade se propõe a estudar o mesmo problema utilizando as mesmas métricas, simuladores e cenários. Como exemplo, temos *RoboCup Soccer* (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009), *RoboCup Rescue* (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009) e *TAC* (Wellman e Wurman, 1999; Vetsikas e Selman, 2002; Cai et al., 2009; Niu et al., 2009), os quais serão apresentados em mais detalhes na seção 2.4.

Esse tipo de iniciativa representa uma tradição na concepção de testes e na forma pela qual pesquisadores procuram avançar o conhecimento. Nesse sentido, o primeiro benchmark foi o *Jogo da Imitação* (Turing, 1950), o qual ainda é utilizado até hoje, mas sem o mesmo empenho. Apesar dos esforços em resolvê-lo terem contribuído para áreas tais como Processamento de Linguagem Natural (PLN), Sistemas Cognitivos, Representação do Conhecimento e Raciocínio Automático, muitas críticas contrárias à

sua adoção foram feitas, sobretudo quanto à sua capacidade limitada de avaliar comportamentos inteligentes e sobre o fato de que o conceito de inteligência empregado é controversa (Hayes e Ford, 1995; French, 2000; Cohen 2005).

Em seguida, foi proposto o uso do *Xadrez* como cenário para estudo de soluções inteligentes (Levinson et al, 1991; Hsu, 2002), o que gerou uma grande quantidade de trabalhos e contribuiu para o desenvolvimento de várias soluções, principalmente em problemas de busca. No entanto, desde a derrota de Kasparov para o *DeepBlue* (Newborn, 2000), este desafio deixou de receber a mesma atenção da comunidade.

2.3 Tipos de Experimentos usando Benchmarks

Benchmarks permitem o estudo comparativo de soluções em um ou mais cenários para melhor compreender o comportamento de cada uma delas. A combinação entre estes dois aspectos (*cenário* e *solução*) possibilita realizar vários tipos de experimentos e obter diversas deduções acerca das habilidades e limitações dos agentes.

Na figura 2.2 são mostradas características de cada experimento, de acordo com a quantidade de soluções e cenários testados. Em cada uma delas, observamos que os parâmetros *multiplicidade de variáveis*, *finalidade* e *tipo de conclusão* podem variar de acordo com o tipo de estudo.

- **Multiplicidade de variáveis** descreve as *possibilidades* de combinação de variáveis entre problema e solução em um experimento, demandando maior ou menor esforço por parte do pesquisador ao projetá-lo e compreendê-lo. Pode ser *alta* ou *baixa*;
- **Finalidade do experimento** descreve a motivação associada ao trabalho. Pode ser quanto ao *aprendizado* ou à *comparação*. No primeiro caso, o pesquisador deseja compreender melhor o comportamento de sua solução mediante determinadas condições do ambiente. No outro, ele deseja comparar o comportamento de sua solução com a de outros pesquisadores;
- **Tipo de conclusão** descreve o quanto é possível generalizar as conclusões no experimento. Pode ser *local* ou *geral*. No primeiro caso, as conclusões se restringem apenas à solução e ao cenário em questão, enquanto no segundo as conclusões podem ser generalizadas a várias soluções ou a vários cenários.

	UM CENÁRIO	VÁRIOS CENÁRIOS
UMA SOLUÇÃO	Baixa Aprendizado Local	Baixa Aprendizado Geral
VÁRIAS SOLUÇÕES	Baixa Competição Geral	Alta Competição Geral

Figura 2.2: Relação entre soluções e cenários na realização de experimentos.

Nos próximos parágrafos iremos analisar cada uma das situações possíveis apresentadas na figura 2.2. Inicialmente, temos o caso em que o pesquisador tem uma *única solução* e irá testá-la em um *único cenário*. Esse tipo de situação é típica em áreas de pesquisa em estágio inicial. A multiplicidade das variáveis pode ser considerada baixa, pois há poucas variáveis envolvidas. O objetivo é o aprendizado, visto que há apenas uma solução. Basicamente o que o pesquisador faz é analisar o comportamento da solução naquele contexto. Daí, todas suas conclusões se limitam a ele, não sendo possível fazer generalizações. Suas conclusões são, portanto, locais.

Posteriormente, temos o caso em que uma *única solução* será testada em *vários cenários*. Nesse caso, é provável que já tenham sido feitos experimentos em um único cenário e se deseja expandir sua compreensão e/ou aplicabilidade de sua solução. Para isso, é preciso analisar como os agentes se comportam em cenários diferentes. A multiplicidade do experimento continua sendo baixa, apesar de ser maior do que no caso anterior. O objetivo do experimento é o aprendizado, pois o pesquisador deseja analisar o comportamento e desempenho de sua solução em diferentes cenários. Por fim, ele pode expandir suas conclusões a respeito das capacidades dos agentes, visto que agora ele pode observar o comportamento do agente em várias situações.

Em seguida, temos o caso em que *várias soluções* são testadas em um *único cenário*. Portanto, temos um experimento em que a multiplicidade é baixa, a finalidade é a competição e o tipo de conclusão é geral. Essa situação ocorre quando se deseja comparar o desempenho de várias soluções entre si. Portanto, as conclusões são restritas a um único cenário.

Finalmente, temos o caso em que *várias soluções* são testadas em *vários cenários*. O experimento consiste em identificar qual solução tem melhor desempenho em cada cenário. Isso é importante porque algumas soluções tendem a ter melhores resultados em alguns cenários, e piores em outros. A multiplicidade das variáveis é alta,

pois envolverá muitas alternativas e combinações entre elas. O objetivo do experimento é a comparação, pois todas as soluções serão comparadas com as demais em todos os cenários. O tipo de conclusão é geral, ou seja, dada a variedade de soluções e cenários, é possível generalizar as conclusões em relação a todos os concorrentes e contextos testados.

Deve ficar claro que a figura 2.2 não é exaustiva e que outras informações poderiam ser obtidas. De toda forma, ela permite apresentar um esclarecimento a respeito dos tipos de experimentos e guiar pesquisadores sobre qual deve ser a melhor alternativa em cada estágio de pesquisa.

Como forma de compreender o estado atual da pesquisa em benchmarks em SMA, na próxima seção iremos apresentar os principais trabalhos desenvolvidos.

2.4 Exemplos de Benchmarks em SMA

O objetivo desta seção é analisar os principais benchmarks desenvolvidos para o estudo de SMA. Primeiro, iremos considerar o *Robocup Soccer* (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009), um benchmark bastante utilizado pela comunidade de SMA e de Robótica. Segundo, iremos analisar o *Robocup Rescue* (Kitano et al., 1999; Takahashi, 2009), um benchmark desenvolvido no contexto de gestão de desastres. Por fim, iremos analisar o *Trading Agent Competition* (Wellman e Wurman, 1999; Vetsikas e Selman, 2002; Cai et al., 2009; Niu et al., 2009), um benchmark para negociação em mercados *on-line*.

2.4.1 Robocup Soccer

A Copa Mundial de Futebol de Robôs, ou Robocup, é uma tentativa de promover a pesquisa em IA e robótica inteligente por meio de um problema padrão onde uma grande variedade de tecnologias podem ser integradas e examinadas (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009). Para que cada jogador (robô físico ou agente de software) seja capaz de jogar futebol no mesmo nível de jogadores humanos profissionais, uma quantidade significativa de técnicas deve ser integrada e vários desafios devem ser superados. Exemplos de áreas pertinentes incluem agentes autônomos, aquisição de estratégias, raciocínio em tempo real e fusão de sensores (Kitano, et al., 1997; Boer e Kolk, 2002).

O principal objetivo da *Robocup* é o desenvolvimento de uma equipe de robôs capaz de vencer a melhor equipe de jogadores do mundo, ou seja, a seleção campeã da Copa de Futebol Mundial, organizada pela FIFA, por volta de 2050 (Kitano et al., 1996). A primeira versão desse campeonato ocorreu em 1996, e dela participaram 35 equipes de 12 países. A versão de 2011 contou com 144 equipes de vários países.

Atualmente, a *RoboCup Soccer* conta com cinco ligas, cada uma com um nível de complexidade diferente (Robocup, 2011):

Uma vantagem da *RoboCup Soccer* é que ela possui um problema de fácil entendimento e que tem um forte apelo, o que atrai a atenção de empresas interessadas em realizar investimentos em pesquisas e de novos talentos. Por outro lado, esse benchmark apresenta problemas tais como a falta de métodos rigorosos na avaliação das técnicas empregadas, dificuldade em aplicar certas técnicas criadas para o problema em aplicações reais e excesso de complexidade do desafio imposto.

2.4.2 RoboCup Rescue

Gestão de desastres é uma das questões sociais mais sérias e pode envolver a ação de uma grande quantidade de agentes heterogêneos em um ambiente hostil (RoboCup, 2011). A principal motivação para o desenvolvimento do *RoboCup Rescue* foi o terremoto de Kobe, que afetou mais de um milhão de pessoas.

A partir da experiência com esse desastre, foi proposto o desenvolvimento de sistemas de informações que apoiassem atividades como coleta de informação, planejamento, tomada de decisão, resgate e combate a incêndios como forma de mitigar prejuízos. Fazendo uso da larga divulgação e do conhecimento adquirido com a *RoboCup Soccer*, foi proposta a *RoboCup Rescue* (Kitano et al., 1999). O objetivo é que esse benchmark possa auxiliar o desenvolvimento e avaliação de sistemas que contribuam com administradores públicos em atividades de gestão de crise, principalmente no que diz respeito à tomada de decisão (Takahashi, 2009).

RoboCup Rescue é uma competição anual e consiste de duas ligas: *simulação* e *robótica*. A liga de simulação foca em planejamento estratégico e coordenação de equipes, enquanto o foco da liga de robôs lida com a capacidade de lidar com operações de resgate e como eles colaboram para realizar determinadas tarefas.

2.4.3 TAC

Com a consolidação da Internet como plataforma de negócios no início da década de 2000, houve naturalmente um crescimento das oportunidades no comércio eletrônico. O surgimento das transações B2B (*bussiness-to-bussiness*) e B2C (*bussiness-to-consumer*) demandou um novo foco de pesquisa: a junção de técnicas baseadas em SMA e comércio eletrônico.

Motivado pela iniciativa de avançar a pesquisa em um domínio de problema por meio de competições, **TAC** (*Trading Agent Competition*) foi proposta por Wellman e Wurman (1999) com o objetivo de prover um benchmark no domínio dos mercados *on-*

line (Wellman e Wurman, 1999; Vetsikas e Selman, 2002; Woodridge et al., 2002; Stone, 2003).

Na versão original, cada agente de software representa um agente de viagem responsável por encontrar as melhores oportunidades de vôos, hospedagem, e tipos de atrações de entretenimento para seus clientes, tanto em termos de preço como de horários e preferência por certos tipos de atrações. O agente deve conciliar os interesses dos seus clientes juntamente com suas possibilidades de pagamento e as oportunidades que surgem, oferecidas pelos vendedores (Wellman e Wurman, 1999). Em outras palavras, TAC requer agentes autônomos em cenários de compra e venda. Cada agente participa de diversos leilões simultaneamente, o que torna a tarefa significativamente complexa. Portanto, a principal habilidade a ser estudada neste benchmark é a negociação.

Nos anos seguintes surgiram outras versões do **TAC**. **TAC-SCM** (*Supply Chain Management*) – Gestão de Cadeia de Fornecedores – tem como objetivo o desenvolvimento de agentes que simulam pequenos fabricantes, que devem competir entre si por fornecedores de matéria prima e clientes, e gerenciar inventários e facilidades de produção (Collins et al., 2006). O objetivo de **CAT**¹ é o inverso de **TAC**: os organizadores da competição disponibilizam agentes corretores e os competidores devem desenvolver regras que combinem compradores e vendedores bem como os valores das comissões dadas aos agentes por realizarem as transações (Cai et al., 2009; Niu et al., 2009). **TAC-AA** (*Ad Auctions*) é voltado para o contexto dos anúncios publicitários associados a resultados de pesquisas em mecanismos de busca. Desenvolvedores devem projetar agentes que simulem a participação em disputas por meio de leilões por espaço quando determinadas palavras chaves forem pesquisadas. Agentes são avaliados de acordo com o lucro em venda de anúncios, ou seja na medida em que tiverem mais sucesso de aparecer nas páginas e forem visitados pelos usuários (Jordan et al., 2010). Mais recentemente surgiu **Power TAC**, cujo objetivo é o desenvolvimento de agentes que simulam o comportamento de corretores de energia elétrica, que devem competir pela produção e consumo de energia, e gerenciar seus portfólios (Ketter et al., 2010; Ketter et al., 2011).

Apesar do uso crescente dos benchmarks mencionados na condução de pesquisas em SMA e robótica, alguns autores tem apontado diversos problemas, o que compromete sua adoção e consequentemente a validade dos seus resultados (REF). Na próxima seção iremos descrever alguns requisitos aos quais um benchmark deve satisfazer.

¹ Segundo seus desenvolvedores, CAT não é apenas o inverso de TAC (para representar a idéia de que seu objetivo é o inverso do TAC), mas diz respeito à palavra CATallics, que representa a ciência dos negócios.

2.5 Comentários Sobre o Capítulo

Este capítulo apresentou o problema de avaliar o progresso da pesquisa em SMA por meio do uso de benchmarks. Nossos objetivos principais foram o de contextualizar a pesquisa de benchmarks em SMA e apresentar demandas que justifiquem seu desenvolvimento. Vimos que benchmarks são fundamentais para a compreensão e avaliação empírica de teorias e técnicas, permitindo dessa maneira analisar de forma criteriosa o progresso da área.

De acordo com o que foi apresentado, o estudo de SMA apresenta uma série de limitações do ponto de vista de sua experimentação, o que compromete a adoção em larga escala de suas soluções. Por um lado, grande parte dos estudos realizados não faz uso de um problema comum ou não há o devido compartilhamento de ferramentas, métricas e cenários. Como consequência aspectos fundamentais do ponto de vista do método científico, tais como replicação de experimentos e comparabilidade de resultados, ficam comprometidos. Por outro, benchmarks existentes são voltados para problemas bastante específicos, o que limita a exploração de técnicas existentes, ou possuem métricas subjetivas acerca do comportamento dos agentes.

O desenvolvimento de benchmarks é uma tarefa complexa que envolve várias atividades e que deve atender a uma série de requisitos a fim de ser útil à comunidade de pesquisadores. Sendo assim, benchmarks devem ser criados utilizando algum processo, metodologia ou diretriz, a fim de facilitar o atendimento às demandas existentes em termos de experimentos e validação científica.

No próximo capítulo será apresentado, um processo com a finalidade de apoiar a criação do benchmark proposto nessa tese. Nos capítulos seguintes, cada uma das atividades será descrita em detalhes.

Capítulo 3

Um Processo para Desenvolvimento de Benchmarks

Conforme visto no capítulo anterior, benchmarks representam uma alternativa importante na condução de experimentos em diversas áreas da Computação, sobretudo em SMA. No entanto, apesar dessa constatação, não conseguimos identificar na literatura processos, métodos, diretrizes ou mesmo estudos sistemáticos que auxiliem pesquisadores na criação de novos benchmarks. Por um lado, isso faz com que sejam adotadas práticas informais, o que compromete a validade dos resultados obtidos nos experimentos, além de dificultar o compartilhamento de esforços, inviabilizando o reuso de conhecimento. Nesse sentido, o objetivo desse capítulo é propor um processo de desenvolvimento de benchmarks denominado de PROBENCH¹.

3.1 Justificativa

A análise comparativa de teorias por meio de experimentos empíricos é uma das atividades mais importantes na metodologia da pesquisa científica. Através dela, é possível verificar se há avanços no estado da arte da área do conhecimento. Para isso, é preciso que os pesquisadores da área em questão adotem as mesmas condições e aparatos experimentais. Dentro do contexto de SC, faz-se necessário identificar a solução que possui melhor desempenho. Estudos comparativos permitem fazer perguntas cujas respostas possibilitam uma melhor compreensão do comportamento das soluções desenvolvidas, como por exemplo: Qual solução atende aos requisitos de desempenho? Como otimizar uma solução? Por que uma solução se comporta de determinada forma sob certas circunstâncias?

Um outro aspecto de fundamental importância na metodologia da pesquisa científica diz respeito à replicação de experimentos. Se os experimentos forem realizados utilizando benchmarks, é natural supor que os demais pesquisadores poderão fazer uso dos mesmos artefatos e das mesmas atividades e, conseqüentemente, será

¹ Do inglês: *Process for Benchmark Development*.

possível replicá-los. Também, será mais simples averiguar a validade dos resultados obtidos e, conseqüentemente, refutar ou confirmar a hipótese em estudo.

Nas últimas décadas, inúmeros trabalhos têm sido desenvolvidos com o objetivo de prover à Ciência da Computação o mesmo rigor científico de outras áreas, como acontece com as ciências naturais. Inspirados pelas abordagens de Melhoria Contínua da Qualidade (MCQ) em Administração (Langley et al., 2009), pesquisadores da Ciência da Computação vêm desenvolvendo benchmarks para os mais diversos tipos de aplicações.

Algumas subáreas de IA, tais como Redes Neurais (Prechelt, 1994) e Web Semântica (Guo et al., 2006; Ferrara et al., 2011), têm se destacado na adoção de benchmarks e apresentam resultados importantes. Apesar de existir um consenso em relação aos benefícios obtidos com a adoção de benchmarks, ela não tem sido adotada com mesmo êxito em outras áreas da IA, como por exemplo, SMA. Abaixo listamos duas possíveis razões para isto:

1. Não foram desenvolvidas abordagens sistemáticas que auxiliem os pesquisadores a conduzir de maneira organizada e criteriosa a criação de benchmarks;
2. Há poucas iniciativas para se promover de forma abrangente e compartilhada o uso dessa prática na área, havendo uma preferência pelo uso da simulação individualizada, sem parâmetros padronizados de referência e com métricas específicas.

Uma questão que merece ser ressaltada é que não existem benchmarks capazes de avaliar todo e qualquer produto ou serviço, pois cada área ou tipo de produto demanda uma forma de avaliação própria, de acordo com seus critérios de desempenho e necessidades dos usuários (Hanks et al., 1993). Mesmo dentro de uma subárea tal como a IA, faz-se necessária a existência de diversos benchmarks a fim de atender às necessidades individuais de cada subárea. Como consequência, é preciso que mais benchmarks sejam desenvolvidos, a fim de apoiar o avanço do estado da arte em cada uma dessas subáreas. Sendo assim, é fundamental que exista um processo sistemático que auxilie os pesquisadores a conduzir o desenvolvimento de benchmarks, informando quais etapas devem ser realizadas e quais artefatos são necessários para a realização de uma avaliação comparativa.

O desenvolvimento de benchmarks deve seguir um processo rigoroso. Além da complexidade inerente ao sistema sendo avaliado e às operações de monitoramento do desempenho, o benchmark deve atender a uma série de requisitos, os quais serão apresentados na seção 4.1.

De acordo com a Norma ISO 9001:2008 (ABNT, 2008), um processo é um conjunto de atividades inter-relacionadas e interatuantes que transformam entradas em saídas. Pressman (2010) afirma que um processo é uma coleção de atividades, ações, e tarefas que são realizadas quando algum produto de um trabalho deve ser criado. Portanto, um processo define, de forma sistemática, a ordem em que as fases devem estar organizadas, a identificação de início e fim de cada uma, quais os atores e quais são seus produtos. Processos devem ser descritos com um nível de detalhamento que possibilite sua adoção de forma clara e simplificada. Assim, durante a concepção de um processo, perguntas referentes a seus componentes básicos devem ser respondidas (Jacobson et al., 2010):

- **Parâmetro:** Quais as entradas de cada fase?
- **Fases:** Quais são as fases que devem ser executadas a fim de criar um benchmark?
- **Papel:** Quem conduz cada fase?
- **Atividade:** Como as atividades de cada fase serão conduzidas? Qual a ordem de execução das atividades? Quando cada uma começa e termina?
- **Produto:** Qual o produto de cada fase?

Sendo assim, um processo permite sistematizar o desenvolvimento de algum produto, método ou serviço a fim de melhorar sua qualidade final e otimizar os recursos envolvidos. No caso específico de benchmarks, a adoção de um processo durante sua criação é particularmente importante, por permitir o reuso de conhecimento e a padronização da documentação.

A realização de um estudo comparativo envolve inúmeras etapas e, dependendo do domínio, pode ser uma tarefa complexa. No entanto, até onde podemos pesquisar, não é comum a adoção de práticas padronizadas de benchmark em computação, sobretudo em alguns domínios da IA, tais como em SMA. Consequentemente, a avaliação dos trabalhos é feita de forma isolada e sempre começando do zero, elevando custos e demandando mais tempo de desenvolvimento. Nesse caso, o **reuso de conhecimento**, por meio da adoção de processos bem definidos e estabelecidos, pode vir a acelerar a criação de benchmarks, reduzindo gastos e tempo.

Adicionalmente, um processo pode auxiliar pesquisadores quanto à **padronização da documentação**, identificando o que é relevante ao relatar um experimento. Neste caso, o que se percebe em grande parte dos trabalhos relatando experimentos é que não há um padrão sobre quais são as atividades de um benchmark que devem ser descritas. Em nosso entendimento, uma das razões para que isso ocorra

diz respeito à falta de um consenso sobre quais são as atividades ou artefatos essenciais a serem adotados pelos pesquisadores.

A fim de promover essa prática, desenvolvemos um processo para criação de benchmarks, denominado de PROBENCH, o qual será descrito na próxima seção.

3.2 PROBENCH

O PROBENCH é um processo para a criação de benchmarks voltados para o estudo de SMA. Seu objetivo é conduzir desenvolvedores através de um conjunto de atividades fundamentais a fim de identificar quais elementos são necessários para avaliar os diversos trabalhos realizados. Do ponto de vista da forma como organizamos o processo, tentamos buscar um equilíbrio entre ter um processo bem definido e controlado, mas ao mesmo tempo primando pela simplicidade e flexibilidade.

Nesta seção são apresentadas as principais fases, atividades e artefatos envolvidos em PROBENCH: *planejamento*, *modelagem do problema*, *definição dos datasets*, *definição das métricas de desempenho* e *desenvolvimento ou escolha de um simulador*.

A figura 3.1 ilustra as fases através de retângulos; por meio de setas são indicados os fluxos de trabalho do processo e, finalmente, algumas atividades e artefatos relacionados a cada uma das fases são enumerados.

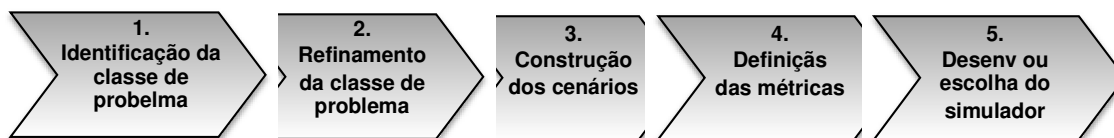


Figura 3.1: Fases de PROBENCH.

3.2.1 Desenvolvimento

O desenvolvimento de PROBENCH foi realizado a partir de uma análise da literatura sobre benchmarks e da experiência de mais de dez anos de pesquisa do grupo de do CIn-UFPE (Machado et al., 2002; Machado et al., 2003; Almeida et al., 2004; Menezes et al., 2006; Moreira 2008; Araújo et al., 2010; Sampaio et al., 2010).

A análise de literatura foi realizada em duas fases. Na primeira, que chamamos de *fase de aprendizado*, a partir de um conjunto reduzido de trabalhos (20 artigos) sobre o uso de benchmarks, identificamos quais artefatos eram abordados em cada artigo. Em seguida, derivamos quais atividades seriam necessárias realizar a fim de produzir cada artefato. Por exemplo, alguns trabalhos mencionam a adoção de cenários de teste. Então, para que isso possa ser alcançado, devemos ter no processo uma etapa que

descreva como cenários de teste devem ser criados. Também identificamos trabalhos que mencionam atividades necessárias para o desenvolvimento de benchmarks, mas que não apresentam detalhes sobre como a atividade deve ser desenvolvida, ou não descrevem o processo como um todo. A partir destes artigos, reunimos quais seriam essas atividades, observando a frequência com que seus respectivos artefatos eram mencionados.

Em seguida, na *fase de avaliação*, a partir de um conjunto maior de trabalhos (80 artigos) sobre o uso de benchmarks, selecionamos um subconjunto de trabalhos representativos utilizando como critérios o intervalo de tempo em que foram publicados (2002 – 2012²) e a abrangência das áreas da Computação tratadas. Em seguida, analisamos quais deles adotavam os artefatos identificados na fase anterior.

O Quadro 3.1 apresenta o conjunto de trabalhos examinados, identificando os aspectos tratados por cada um.

² Uma exceção nesse critério foi o trabalho de Prechelt (1994), devido à sua importância para a compreensão do que são benchmarks e a definição de sua estrutura.

Quadro 3.1: Relação entre trabalhos sobre benchmarks em SC.

	Domínio	Planejamento	Modelagem	Cenários	Métricas	Simulador	TOTAL POR TRABALHO (Porcentagem)
Prechelt, 1994	RN	•		•			□ □
Howe e Dalhman, 2002	PLAN	•		•	•		60
Keog e Kasetty, 2002	MD	•		•	•		60
Eytani et al., 2005	SMT	•	•	•		•	80
Albayraktaroglu et al., 2005	BI	•		•	•	•	80
Raths et al., 2005	PAT	•		•	•		60
Gardiner et al., 2006	RDL					•	20
Wörner e Wörn, 2006	SCP		•	•	•	•	80
Zöller, 2006	SMA	•	•	•	•		80
John, 2006	HW			•	•	•	60
Narayanan et al., 2006	MD	•		•	•	•	80
Guo et al., 2006	WS	•		•	•	•	80
Borghettie e Sodomka, 2006	SMA	•		•	•	•	80
Black e Hougen, 2006	AAC	•				•	40
Gardiner et al., 2006	SLD	•	•	•		•	80
Such et al., 2007	PSMA	•		•	•		60
Dovier et al., 2007	SLD	•		•	•		60
Duchateau et al., 2007	CEX	•		•	•	•	80
Pelánek et al., 2007	CM		•	•		•	60
Rivoire et al., 2007	CE	•		•	•	•	80
Liu et al., 2007	RI			•	•		40
Alexe et al., 2008	ME	•		•	•		60
Sun et al., 2009	RC	•	•		•	•	80
Che et al., 2009	ME				•		20
Giot et al., 2009	SBM	•	•	•	•	•	100
Haindl e Mikês, 2009	PI			•	•	•	60
Pronobis et al., 2010	RI	•	•	•	•		80
Ferrara et al., 2011	WS	•	•	•	•	•	100
Scheireber, 2011	SCP		•				20
TOTAL POR FASE (Porcentagem)	--	70,0	33,3	83,3	76,7	60,0	--

Legenda: RN (Redes Neurais), WS (Web Semântica), RI (Recuperação da Informação), PI (Processamento de Imagem), SBM (Sistemas de Biometria), ME (Mapeamento de Esquemas), RC (Redes de Computadores), CE (Consumo de Energia em Sistemas Computacionais), CM (Checagem de Modelos), CEX (Casamento de Esquemas XML), SLD (Sistema de Linguagem Declarativa), PSMA (Plataforma de Sistemas Multiagente), AAC (Agentes de Aprendizagem Coletiva), SMA (Sistemas Multiagente), MD (Mineração de Dados), HW (Hardware), PLAN (Planejamento), SCP (Sistemas de Controle de Produção).

Ao final da análise dos trabalhos, concluímos que algumas etapas são mais mencionadas do que outras, evidenciando uma falta de padronização em relação à forma como um benchmarking deve ser conduzido. Por exemplo, enquanto que *cenários* são considerados em 83%, *modelagem* é considerada em apenas 33,3%. Em relação às demais etapas, o mesmo também ocorre. Essa forma desorganizada de se criar, utilizar e adotar benchmarks se deve aos seguintes fatores:

- *Abordagem:* a forma como os benchmarks são desenvolvidos não leva em consideração um conjunto padrão de atividades e artefatos. Sendo assim cada autor (ou grupo de autores) adota um conjunto particular de atividades, o que prejudica a divulgação e replicação de resultados;
- *Documentação:* não há um consenso a respeito de quais informações acerca das atividades e/ou artefatos devem ser mencionados em um artigo científico. Esse fato impede que ele possa ser analisado adequadamente e os experimentos apresentados sejam replicados.

Consideramos que todas as atividades abordadas nessa análise são importantes, e que, conseqüentemente, atividades referentes a cada um devem fazer parte de PROBENCH.

3.2.2 Visão Geral

PROBENCH é formado por cinco fases: (1) identificação da classe de problema, (2) detalhamento da classe de problema, (3) construção de cenários, (4) definição das métricas e (5) seleção ou implementação do simulador. Cada fase é responsável por conduzir os desenvolvedores na construção de uma parte específica do benchmark, desde a coleta das informações iniciais relacionadas à sua demanda até o desenvolvimento ou identificação das ferramentas de apoio. Conforme pode ser visto na figura 3.1, as fases devem seguir uma sequência de tal forma que o produto de uma seja utilizado na realização da seguinte.

A fase de **Identificação da Classe de Problema** tem como objetivos pesquisar, selecionar e/ou criar uma classe de problema a ser utilizada no benchmark. A fase de **Detalhamento da Classe de Problema** tem como finalidade aperfeiçoar o conhecimento acerca da classe de problema escolhida ou criada na fase anterior a fim de remover inconsistências e suprimir lacunas conceituais. Em seguida, a fase de **Construção de *datasets*** visa criar as instâncias do problema que serão usados nos experimentos. Eles serão obtidos por meio da instanciamento dos modelos descritos na fase anterior. A fase de **Definição das Métricas** tem como objetivo a definição das métricas de desempenho que serão utilizadas para o estudo do sistema. A partir delas será possível comparar um determinado trabalho com outros existentes na literatura ou no mercado. Por fim, a fase de **Desenvolvimento ou Escolha do Simulador** tem como objetivo pesquisar a existência de simuladores que atendam às necessidades do benchmark sendo desenvolvido. Caso já exista, deve se analisar a possibilidade de adotá-lo. Do contrário, um novo simulador deve ser desenvolvido.

Conforme Pressman (2010, p. 40), projetos reais raramente seguem um fluxo sequencial. Sendo assim, consideramos que PROBENCH deve ser tão flexível o quanto possível, deixando a critério dos pesquisadores estabelecerem iterações e retornos às etapas anteriores a fim de acomodar novo conhecimento ou corrigir eventuais falhas detectadas. De forma semelhante, sugerimos técnicas e/ou ferramentas que podem ser utilizadas em cada atividade, mas deixando a critério do desenvolvedor qual será adotada.

A fim de facilitar a utilização do processo, disponibilizamos sempre que necessário uma pergunta-chave junto às tarefas de cada atividade. Essa pergunta sintetiza o que a tarefa deve fazer e guia o desenvolvedor na realização da atividade.

3.2.3 Papéis Chave

O desenvolvimento de um processo pode envolver a participação de diversas pessoas. Para fins de eficiência e organização das atividades é necessário atribuir devidamente as

responsabilidades a cada um dos participantes de acordo com suas habilidades. Para cada responsável damos o nome de **papel chave**. No PROBENCH são recomendados os seguintes papéis:

- **Pesquisador:** é responsável por conduzir o desenvolvimento do benchmark para realizar experimentos a fim de realizar pesquisas. Seu papel é decisivo na concepção do benchmark, pois é a partir de sua percepção e de suas demandas que o processo será realizado. Além disso, ele irá fornecer as informações necessárias para que o benchmark atenda suas demandas;
- **Desenvolvedor:** é o responsável por desenvolver um ou mais artefatos relacionados ao benchmark devido a habilidades específicas. Por exemplo, em função de suas habilidades como programador ele pode ser requisitado para implementar o simulador ou caso disponha de conhecimentos acerca de engenharia de ontologias, criar uma ontologia;
- **Especialista:** atua como consultor em relação a uma determinada área do conhecimento a qual o pesquisador não possui conhecimento profundo, mas que pode ser útil para o estudo de determinadas técnicas. Por exemplo, no caso de um benchmark para Logística de Manufatura (Zöller et al., 2006), um especialista pode fornecer informações acerca dos desafios enfrentados na realização das tarefas dessa área e descrever formas de avaliar o desempenho;
- **Usuário:** é aquele que irá utilizar o benchmark a fim de realizar experimentos científicos usando SMA. Ele participa do desenvolvimento apenas como avaliador dos recursos do benchmark.

É importante ressaltar que um mesmo participante pode desempenhar mais de um papel ao longo do processo. Similarmente, um mesmo papel pode estar presente em mais de uma fase. Por exemplo, o pesquisador está presente em todas as fases e é o responsável pela criação do benchmark. Portanto, ele deve gerenciar todas as atividades, incluindo aquelas que não detalharemos aqui, tais como definir cronograma, avaliar a qualidade dos artefatos, entrevistar o especialista etc. Como forma de simplificar, daqui em diante iremos nos referir ao termo **ator** de forma genérica quando qualquer um ou todos esses papéis puderem realizar alguma atividade.

3.3 Identificação da Classe de Problema

3.3.1 Descrição

A primeira fase durante o desenvolvimento de um benchmark de acordo com o PROBENCH deve ser a *identificação da classe de problema*. Sendo assim, o

desenvolvedor deve identificar ou criar classes de problema que atendam às necessidades de experimentação dos usuários.

Os desenvolvedores poderão sugerir ajustes a qualquer tempo na definição da classe de problema a fim de corrigir eventuais erros ou eventuais limitações identificadas.

3.3.2 Artefatos de Entrada

A realização dessa fase deve partir da *descrição da motivação*, do *objetivo do benchmark* e dos *requisitos do benchmark*. Para isso, é necessário que seja realizado anteriormente um estudo de viabilidade a fim de verificar a real necessidade e/ou eventuais impedimentos para o desenvolvimento do benchmark. A partir dele são desenvolvidos os dois artefatos de entrada. A descrição da motivação deve ressaltar a importância em se avançar o estado da arte em determinada área de pesquisa. Por outro lado, os objetivos do benchmark sem necessariamente abordar os requisitos que ele deve atender ao final.

De posse dos artefatos de entrada, os atores envolvidos devem identificar quais são os critérios de seleção da classe de problema. Isso significa dizer que para o estudo de uma determinada área do conhecimento podem existir várias classes de problema. É preciso, portanto, escolher criteriosamente aquela que se adequa melhor. A partir desses critérios será possível identificar quais classes de problema podem ser utilizadas no benchmark. É possível que os desenvolvedores já disponham de um ou mais problemas que são utilizados com frequência pela comunidade de pesquisadores e que atendem a esses critérios. Nesse caso, essa fase será bastante simplificada. Do contrário, será feita uma pesquisa a fim de identificar problemas que possuem interessantes de serem solucionados. Uma vez tendo escolhido a classe de problema, será elaborada uma descrição resumida da classe de problema. Nessa fase, é fundamental que os desenvolvedores descrevam de forma clara e consensual o que é o problema.

3.3.3 Atores

Nessa fase do processo os atores são principalmente os **pesquisadores** da área alvo. Isso se deve ao fato de que são eles que rotineiramente lidam com os problemas da área e, conseqüentemente, tem a percepção dos seus problemas e demandas. Eventualmente **especialistas** podem participar com o propósito de opinar sobre a viabilidade do problema escolhido para a realização dos estudos.

3.3.4 Atividades

Esta fase é dividida em cinco atividades: *identificação dos critérios de seleção da classe de problema*, *pesquisa de classes de problemas existentes*, *escolha da classe de problema*, *obtenção da descrição da classe de problema* e *desenvolvimento da descrição da classe de problema* conforme ilustrado na figura 3.2.

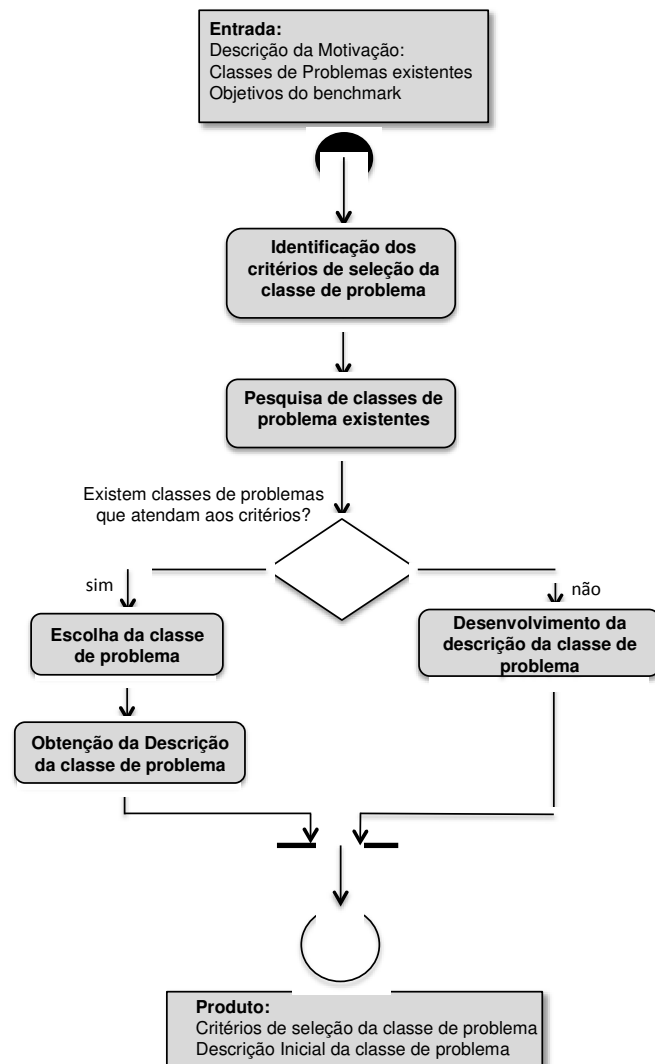


Figura 3.2: Seleção da classe de problema.

Identificação dos critérios de seleção da classe de problema: De posse dos artefatos de entrada, os desenvolvedores devem identificar quais são os critérios de seleção da classe de problema. Isso significa dizer que para o estudo de uma determinada área do conhecimento podem existir várias classes de problema. Além disso, é fundamental

escolher aquela que se adequa melhor aos interesses de pesquisa de uma determinada comunidade. Por exemplo, como forma de possibilitar a comparação de métodos de negociação em SMA, conforme visto na subseção 2.4.3 TAC, Wellman e Wurman (1999) propuseram como classe de problema do benchmark TAC a realização de leilão de passagens, diárias em hotéis e ingressos para passeios turísticos. A partir dos critérios será possível, portanto, identificar qual classe de problema deverá ser utilizada no benchmark. É possível que os desenvolvedores já disponham de um ou mais problemas que são utilizados com frequência pela comunidade de pesquisadores e que atendam total ou parcialmente a esses critérios. No primeiro caso, essa fase será bastante simplificada, pois a atividade de *pesquisa da classe de problema* não será realizada. Caso contrário, é preciso analisar se é necessário fazer ajustes na classe de problema a fim de satisfazer os critérios não atendidos ou se esses não são essenciais em face do objetivo do benchmark.

P.: *Quais são os critérios que devem ser utilizados para selecionar uma classe de problema?*

Pesquisa de classes de problema existentes: Caso os desenvolvedores não disponham a priori de uma classe de problema, deve ser feita uma pesquisa a fim de identificar se existe alguma que atenda às necessidades da comunidade de pesquisadores. A forma como essa pesquisa é realizada deve ser tão variada quanto possível, indo desde o uso da própria experiência em trabalhar com o problema até aos especialistas que podem ser consultados. No caso de RoboCup Soccer, por exemplo, os projetistas tiveram que consultar manuais contendo as regras do esporte (Kitano et al., 1996) a fim de utilizá-lo adequadamente. A pesquisa de classes de problema deve levar em consideração os critérios identificados no início dessa atividade. Dessa forma, os desenvolvedores devem identificar situações no mundo real em que essas habilidades ou serviços sejam explorados.

P.: *Existe alguma classe de problema que possa ser utilizada no benchmark?*

Escolha da Classe de Problema: Uma vez que o desenvolvedor identificou uma ou mais classes que a princípio podem ser usadas no benchmark, ele pode se concentrar nelas e realizar uma análise minuciosa. Para isso ele deve selecionar aquela que melhor atender aos critérios que foram elencados na atividade “*Identificação dos critérios de seleção da classe de problema*”.

P: *Qual a classe de problema adequada para o benchmark?*

Obtenção da descrição da classe de problema: Após os desenvolvedores identificarem a classe de problema que será adotada no benchmark, deve-se obter sua descrição inicial, a qual representa de forma consensual o entendimento de todos os

participantes. A importância dessa descrição se deve ao fato de que um determinado problema pode ter variações no mundo real ou que os desenvolvedores podem ter interpretações diferentes. Também pode ocorrer de o problema ser amplo ou complexo demais para ser estudado inteiramente em uma primeira versão do benchmark.

Desenvolvimento de classe de problema: No caso de não existir uma classe de problema apropriada para o benchmark a ser criado, os desenvolvedores devem propor uma nova e elaborar uma descrição inicial. O exemplo de uma situação típica em que isso ocorre é quanto à adoção de *toy examples*, o que é uma prática bastante comum em determinadas áreas. Norvig e Russel (2004), por exemplo, sugerem o *mundo de wumpus* como uma forma de estudar o comportamento de agentes inteligentes. Apesar de que a classe de problema não corresponde a um problema do mundo real, ela deve permitir que novas técnicas possam ser estudadas em menor escala e em seguida ser utilizadas no desenvolvimento de soluções reais.

3.3.5 Técnica Sugerida para Execução da Fase

A fase de *seleção da classe de problema* demanda conhecimento tanto do domínio para o qual o benchmark será criado como daqueles relacionados com os cenários de experimentação. Isso decorre do fato de que mesmo que o benchmark tenha como objetivo a realização de experimentos em uma área, a classe de problema pode não ter relação direta com a área de conhecimento dos pesquisadores. No caso do benchmark para Redes Neurais PROBEN1, por exemplo, são fornecidos cenários relacionados a problemas tais como diagnose de câncer de mama, diabetes e limites de crédito para empréstimo (Prechelt, 1994). Dessa forma, pesquisadores da área de Redes Neurais devem ter algum conhecimento sobre biologia ou mercado de crédito. Uma maneira de obter esse conhecimento é naturalmente através da literatura. Por isso sugerimos a adoção de técnicas de **revisão sistemática da literatura** (Kitchenham, 2004; Biolchini et al., 2005). Uma outra é dispor de especialistas na equipe de desenvolvimento, os quais serão consultados para fornecer informações sempre que necessário.

Em adição, sugerimos a adoção de técnicas de Engenharia de Requisitos (Hull et al., 2004; Pressman, 2010) durante a **identificação de critérios**. No entanto, consideramos que os pesquisadores devem ponderar a complexidade dessas técnicas, sobretudo se os desenvolvedores não tiverem experiência na sua utilização, e do problema que serão abordados no benchmark. Nesse caso, consideramos que uma lista elaborada com uma breve descrição de cada critério pode ser o suficiente.

3.3.6 Produto

Os artefatos produzidos nessa etapa são: *critérios de seleção da classe de problema* e *descrição inicial da classe de problema*.

3.4 Refinamento da Classe de Problema

3.4.1 Descrição

Essa etapa visa auxiliar desenvolvedores a compreender melhor a classe de problema. Para isso ela deve permitir a obtenção de um conhecimento mais aprofundado acerca do domínio do problema, e a sua consequente modelagem das entidades que fazem parte do problema em questão. No primeiro caso os desenvolvedores devem estender o conhecimento acerca da classe de problema além da descrição inicial obtida na fase anterior. No segundo caso, a classe de problema deve ser modelada utilizando alguma notação gráfica tal como UML (Booch et al., 2005).

Uma vantagem de se modelar um determinado domínio de problema é simplificar sua compreensão. Dessa forma, projetistas podem se concentrar no que há de mais importante e ignorar o resto. A modelagem permite identificar, dentre outras coisas, os elementos que compõem o vocabulário de uma área do conhecimento e como eles estão interligados, facilitando seu entendimento. Este corpo de conhecimento servirá como uma referência comum para comunicação entre desenvolvedores e entre estes e os pesquisadores que irão utilizar o benchmark. Outra vantagem é que o modelo também serve como referência básica para instanciar os elementos do problema e compor diferentes cenários de teste. Portanto, o modelo será utilizado também na fase de construção dos cenários conforme veremos no capítulo 6. Cada instância do modelo descreve um cenário por meio da qual se deseja analisar o desempenho do sistema.

3.4.2 Entrada

O artefato de entrada da fase de *refinamento da classe de problema* é a descrição inicial da classe de problema obtida na fase de *seleção da classe de problema*.

3.4.3 Papéis Chave

Os papéis chave da fase de modelagem de problema são **pesquisadores**, **desenvolvedores do benchmark**, e **especialistas** em áreas afins ao benchmark. No entanto, dependendo da complexidade do problema e do conhecimento dos pesquisadores acerca do mesmo, o desenvolvedor e o especialista podem ser opcionais.

3.4.4 Atividades

Essa fase possui quatro atividades conforme ilustrado na Figura 3.3: Refinamento da classe de problema.

análise detalhada da classe de problema, aperfeiçoamento da classe de problema, modelagem da classe de problema, e validação do modelo da classe de problema.

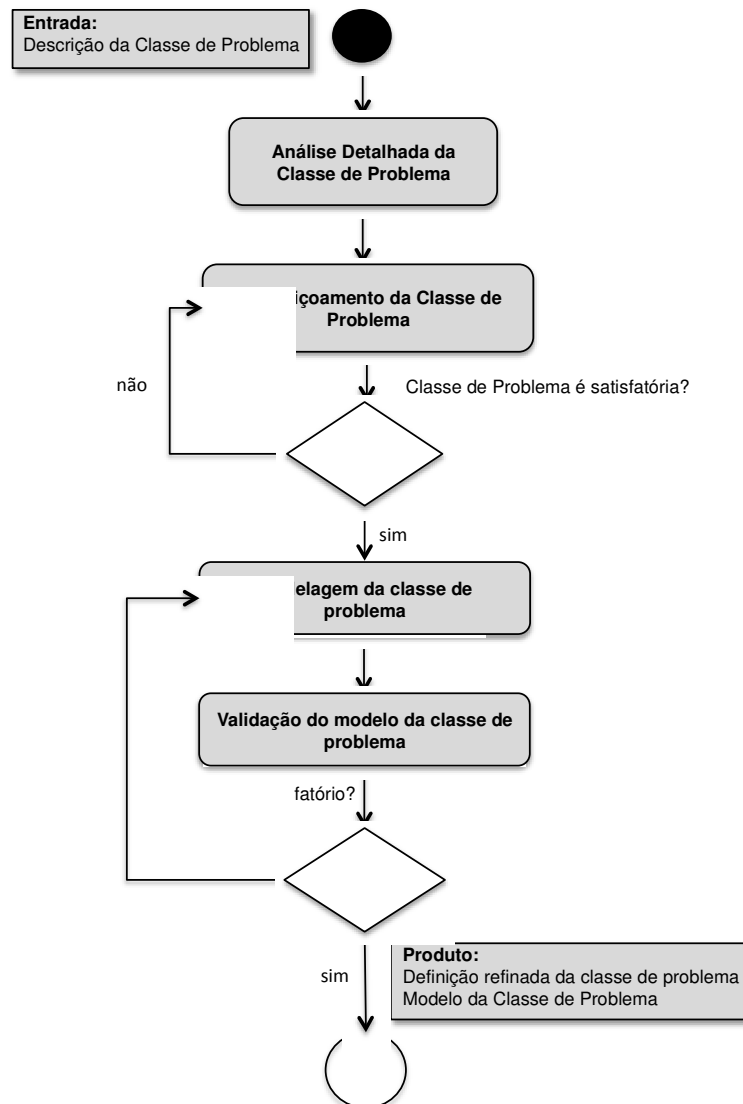


Figura 3.3: Refinamento da classe de problema.

Análise Detalhada da classe de problema: consiste em analisar a classe de problema a fim de aprofundar o conhecimento a respeito bem como identificar se existem limitações que possam inviabilizar sua adoção no benchmark a ser desenvolvido. Em

caso positivo, essas limitações devem ser elencadas a fim de que os atores possam sugerir formas de superá-las.

P.: *Quais são as limitações da classe de problema?*

Aperfeiçoamento da classe de problema: o sucesso da criação de um benchmark depende sobretudo da classe do problema. É fundamental, portanto, que os desenvolvedores se empenhem em obter uma definição a mais concisa possível, além de conhecer mais detalhes e preencher eventuais lacunas. Para isso eles devem obter a descrição da classe de problema de forma iterativa, em um processo conjunto de discussão e aperfeiçoamento. De acordo com a Figura 3.3: Refinamento da classe de problema.

o resultado da atividade aperfeiçoamento da classe de problema deve ser analisado a fim de identificar se ele é satisfatório ou não. Em caso negativo, os desenvolvedores devem refiná-la até o ponto em que ela for considerada consensualmente como tal. Nessa atividade, a literatura deve ser escrutinada e especialistas devem ser consultados caso não façam parte da equipe de desenvolvimento.

P.: *A classe de problema é satisfatória?*

Modelagem da classe de problema: uma vez que uma definição satisfatória do problema seja obtida, os desenvolvedores devem voltar seus esforços à concepção de um modelo que represente a classe de problema. Essa atividade consiste em transformar as informações reunidas na atividade anterior em uma representação gráfica. Existem diversas formas de se fazer isso, mas no geral devem ser ilustradas as entidades da classe de problema, seus atributos e os relacionamentos que existem entre elas. Também é importante descrever a cardinalidade ou quantidade de instâncias de cada entidade. Adicionalmente ao tipo de modelo conceitual escolhido (ontologia, taxonomia, diagrama de classe etc.), o qual captura o conhecimento estático da classe de problema, pode ser interessante criar modelos que capturem seu comportamento ou aspectos dinâmicos, como por exemplo, diagrama de estados, fluxograma, rede de petri etc.

Validação do modelo da classe de problema: o resultado da modelagem deve ser constantemente analisado a fim de identificar inconsistências. Sendo assim, mesmo que haja esse cuidado durante a *modelagem da classe de problema*, é importante que após sua finalização o modelo seja validado. Isso é importante, sobretudo, quando os membros da equipe não possuem um conhecimento aprofundado acerca do domínio de problema, sendo necessário consultar um especialista quando o modelo estiver pronto a fim de consolidá-lo. Sendo assim, deve-se observar se o modelo desenvolvido é

coerente com a descrição do domínio do problema levantada na fase anterior e com o conhecimento existente acerca do assunto.

P.: *O modelo é satisfatório?*

3.4.5 Técnica Sugerida

Uma vez que as atividades dessa fase giram em torno da concepção de um modelo do problema, as técnicas sugeridas são aquelas disponíveis em diferentes áreas da Ciência da Computação tais como Engenharia de Software, Sistemas de Gerenciamento de Banco de Dados e Representação do Conhecimento: *modelagem orientada a objetos* (Larman, 2004), *modelagem de bancos de dados* (Teorey et al., 2006), e *engenharia de ontologias* (Gómez-Pérez et al., 2010), respectivamente. A escolha da abordagem deve levar em consideração o nível de detalhamento necessário para descrever as entidades e o conhecimento do modelo escolhido por parte dos atores.

3.4.6 Artefato Produzido

Os artefatos produzidos na fase de *detalhamento da classe de problema* devem ser capazes de fornecer um entendimento claro sobre o que é o problema adotado no benchmark. Eles consistem da descrição do problema por meio textual e do modelo de alto nível.

3.5 Construção dos Datasets

3.5.1 Descrição

Nessa fase os atores devem construir os *datasets* que serão utilizados nos experimentos. Para isso eles devem instanciar as entidades do modelo da classe de problema através da atribuição de valores às suas variáveis.

3.5.2 Artefatos de Entrada

Os artefatos necessários para a realização dessa fase são os seguintes: *definição detalhada da classe de problema* e *modelo da classe de problema*. A importância do primeiro se deve ao fato que é preciso compreender satisfatoriamente a classe de problema a fim de construir os *datasets*. Em relação ao segundo, os cenários serão criados a partir da instanciamento das entidades do modelo tendo como base as variáveis principais de cada uma.

3.5.4 Papéis

Nessa fase teremos os seguintes papéis envolvidos na criação do benchmark: **pesquisador** e **especialista**. O pesquisador deve indicar quais atributos são relevantes no modelo a fim de compor os cenários de teste. Em relação ao especialista, dependendo da complexidade e do conhecimento do pesquisador acerca do problema do benchmark, a participação do pesquisador pode ser opcional.

3.5.5 Atividades

Nessa fase teremos as seguintes atividades conforme a figura 3.4: *descrição dos cenários de interesse, identificação dos atributos primários e secundários, identificação das restrições, pesquisa restrições de problema, pesquisa literatura, definição de valor, armazenamento de valor e instanciação de cenários*.

Descrição de cenários de interesse: essa atividade identifica aqueles aspectos da classe de problema que influenciam o desempenho das soluções analisadas no benchmark em relação a algum objetivo de estudo. Por exemplo, no trabalho de Almeida e colegas (2003) foi observado de que forma o grafo influencia o desempenho dos agentes, tanto no que diz respeito ao **tipo** quanto ao **tamanho**. Outro tipo de aspecto do domínio de problema que afeta o desempenho dos agentes em um problema da patrulha é a *comunicação entre os agentes*.

P.: *Quais aspectos do domínio de problema afetam o comportamento da solução?*

Identificação dos atributos primários e secundários: atributo primário é todo aquele que pode influenciar o comportamento da solução. Por isso é importante variar os valores dos atributos primários em cada cenário a fim de observar como o desempenho dos agentes é afetado. No exemplo citado na atividade anterior, ao estudar a influência de grafos no desempenho de determinados agentes, *tipo* e *tamanho* do grafo são atributos primários. Por outro lado, os atributos secundários são aqueles que não exercem influência no desempenho dos agentes em relação a um objetivo de estudo. Sendo assim, seus valores são mantidos fixos em todos os cenários.

P.: *Quais são os atributos primários e secundários do domínio de problema para um determinado objetivo de estudo?*

Identificação das restrições: a escolha dos atributos pode ser restringida por três motivos durante a elaboração dos cenários: *restrição do problema*, *restrição por decisão de projeto*, e *restrição por valor default*. No primeiro caso, devem ser adotados para determinados atributos, valores bem definidos pelo problema, não devendo os

desenvolvedores adotarem valores diferentes sob pena de tornar a solução inviável do ponto de vista prático ou prejudicar a comparabilidade. No segundo caso, para fins de simplificação do benchmark, os desenvolvedores adotam valores básicos para determinados atributos. Por último, na *restrição por valor default* os valores de determinados atributos são aqueles largamente adotados na literatura. Uma vez tendo sido verificada a existência de restrições, faz-se necessário identificar e detalhar quais são elas e descrever de que forma influenciam no domínio de valores dos atributos primários. Caso contrário, essa atividade não será realizada.

P.: *Existem restrições para os atributos da classe de problema? Quais são as restrições de cada atributo?*

Pesquisar restrição de problema: essa atividade é realizada quando a própria definição do problema restringe determinados atributos a assumirem certos valores na realização dos experimentos. Por exemplo, no caso de uma patrulha militar, os agentes patrulhadores não podem se comunicar sob o risco de suas mensagens serem interceptadas pelos inimigos.

P.: *Qual a disponibilidade de valores para cada um dos atributos?*

Pesquisa literatura: em algumas situações os valores de alguns atributos serão os mesmos de trabalhos já adotados na literatura. Uma das razões para isso é que esses valores tem se mostrado úteis nos trabalhos existentes, não sendo necessário, portanto, utilizar outros diferentes.

P.: *Há valores disponíveis na literatura que possam ser utilizados? Há valores que possam ser informados pelos especialistas?*

Definição de valores: essa restrição pode se dar pela indisponibilidade de valores para algum atributo como pela decisão de projeto tomada pelos atores. Por exemplo, em um estudo de PMA voltado para a vigilância de áreas urbanas o pesquisador não dispõe dos tamanhos das ruas patrulhadas, então ele define valores artificialmente.

P.: *Como definir artificialmente valores para atributos?*

Armazenamento de valores: o valor de cada atributo que for sendo obtido seja por meio da descrição do próprio problema, pela pesquisa na literatura ou pela própria criação, deve ser armazenado temporariamente até que todos os atributos tenham sido analisados e seus valores obtidos. É preciso que se verifique ao final se há algum atributo que não tenha sido analisado ou que não possua valores.

P.: *Há algum atributo sem valor?*

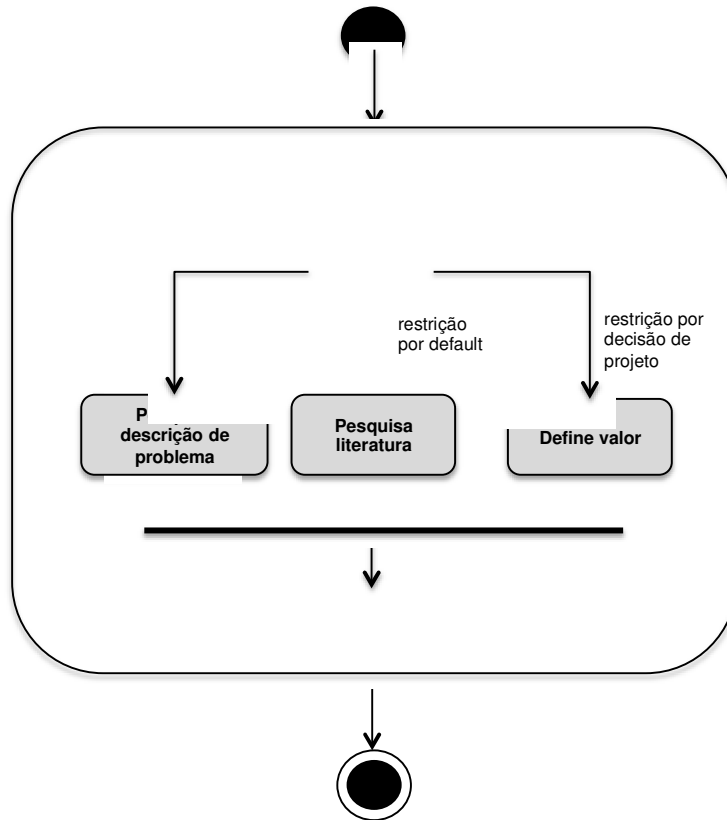


Figura 3.4: Construção de cenários.

Instanciação de Cenários: quando os valores de todos os atributos do critério de desempenho sendo analisado estiverem disponíveis, os cenários serão criados. Isso é feito usando os valores dos atributos (primários e secundários) armazenados. Por meio da variação do valor de um atributo primário a_i enquanto se mantém fixos os valores dos demais atributos primários a_j (tal que $i \neq j$) e dos atributos secundários. Esse processo é repetido para cada atributo primário. Cada combinação formada representa um cenário. Por motivos óbvios, cenários com valores repetidos devem ser descartados. Por exemplo, sejam dois atributos primários x e y , tal que $x : \{1,2\}$ e $y : \{3,5\}$, e dois atributos secundários w e z , tal que $w : \{7\}$ e $z : \{9\}$, então teremos os seguintes cenários: $c_1 = \{1,3,7,9\}$, $c_2 = \{1,5,7,9\}$, $c_3 = \{2,3,7,9\}$ e $c_4 = \{2,5,7,9\}$. Conforme ilustrado na Figura 3.5: Instanciação de cenários., a atividade *Instanciação de Cenário* é formada por um conjunto de sub-atividades.

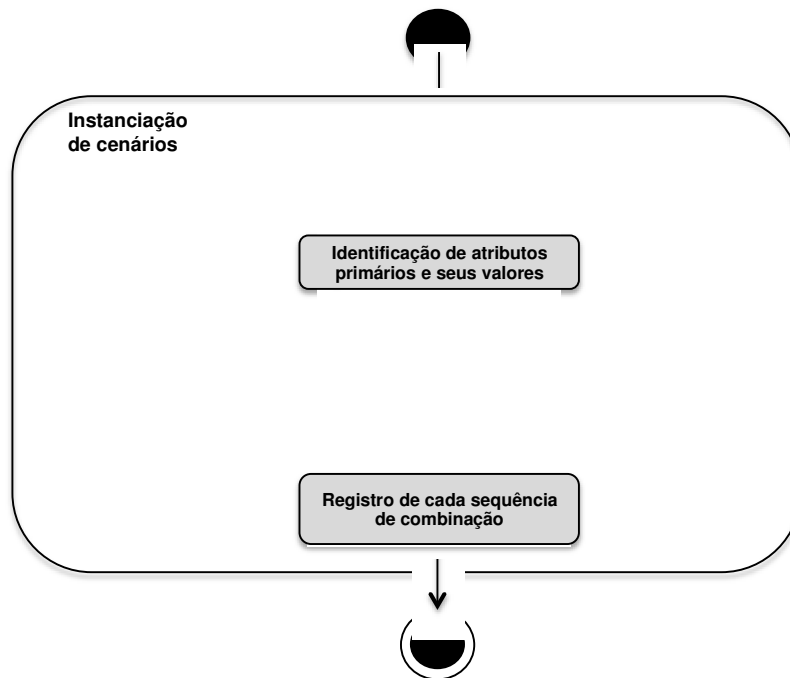


Figura 3.5: Instanciação de cenários.

3.5.5 Técnica Recomendada

Uma vez que a principal atividade dessa fase é a identificação dos atributos (primários e secundários) da classe de problema, consideramos que não há necessidade de nenhuma técnica existente em particular. A seleção dos atributos relevantes e consequente escolha dos valores de interesse devem ser feitas por meio de exemplos em trabalhos existentes ou ser criados pelos pesquisadores.

3.5.6 Artefato Produzido

Os artefatos produzidos na fase de *construção dos cenários* são a *descrição dos atributos* e *os cenários*.

3.6 Definição das Métricas

3.6.1 Descrição

Nessa fase os desenvolvedores devem identificar quais métricas serão utilizadas no benchmark. Para isso os desenvolvedores devem fazer uso da *definição detalhada da classe de problema* e do *modelo da classe de problema*. Através desses artefatos deve-se identificar quais são os serviços oferecidos pelos agentes ou quais são os recursos

que eles consomem. As métricas devem ser elaboradas de forma a permitir que qualquer trabalho naquele campo de pesquisa possa ser avaliado da mesma maneira. Como resultado dessa fase os desenvolvedores devem obter um conjunto de métricas que permita avaliar satisfatoriamente as soluções desenvolvidas.

3.6.2 Artefatos de Entrada

Os artefatos necessários para a realização dessa fase são os seguintes: *definição detalhada da classe de problema* e *modelo da classe de problema*. O primeiro é usado para identificação dos serviços do sistema e dos requisitos e o segundo para identificar quais atributos serão utilizados como métricas de desempenho dos agentes.

3.6.3 Atores

Os atores que participam dessa fase do processo são o *pesquisador* e o *especialista*. O primeiro é responsável por definir quais aspectos do comportamento da solução devem ser avaliados pelo benchmark e o segundo por adicionar outros aspectos que não foram previstos pelos pesquisadores.

3.6.4 Atividades

Essa fase possui as seguintes atividades: *identificação dos serviços do sistema*, *identificação dos requisitos de cada serviço*, *identificação das métricas para cada serviço*, *descrição das métricas* e *revisão das métricas* conforme ilustrado na figura 3.6.

Identificação das tarefas realizadas pelo SMA: os desenvolvedores devem descrever claramente quais são os serviços do sistema a fim de poder compreender qual deve o critério-objetivo a ser usado na quantificação do desempenho. Exemplos de serviços são estabelecer conexões em uma rede de computadores, identificar caracteres escritos manualmente ou reconhecer imagens;

P.: *Quais são as tarefas que os agentes realizam?*

Definição dos requisitos de cada tarefa: a partir dos requisitos é preciso definir os critérios-objetivo das soluções propostas. Em alguns casos, o critério-objetivo descreve a necessidade de maximizar certas variáveis, enquanto em outras deve minimizar. Exemplos de critérios-objetivo relacionados ao estabelecimento à comunicação em uma rede de computadores temos responsividade, utilização de recursos e confiabilidade (Jain, 1991).

P.: *Quais são os requisitos de cada tarefa?*

Identificação das métricas para cada tarefa: métricas devem atender aos requisitos de desempenho observando os critérios-objetivos estabelecidos na atividade *definição dos requisitos de cada serviço*. Por exemplo, para o critério **responsividade** temos *tempo necessário para realizar o estabelecimento de conexão, tempo de resposta e tempo de reação*, **utilização de recursos** (*quantidade de memória alocada*), **confiabilidade** (*quantidade de pacotes de conexões descartados por erro*) etc.

P.: *Quais são as métricas de cada tarefa?*

Descrição das métricas: é essencial que as métricas sejam compreendidas de forma inequívoca. Para isso faz-se necessário apresentar uma descrição textual ou usando notação matemática.

P.: *O que é cada uma das métricas?*

Validação das métricas: métricas devem ser revisadas por todos os desenvolvedores, sobretudo quando a equipe consistir de várias pessoas diferentes ou quando a equipe for dividida em grupos menores, cada um sendo encarregado, por exemplo, por um critério. Para isso, deve-se examinar se cada métrica atender aos critérios que foram estabelecidos no início da atividade.

P.: *As métricas são válidas?*

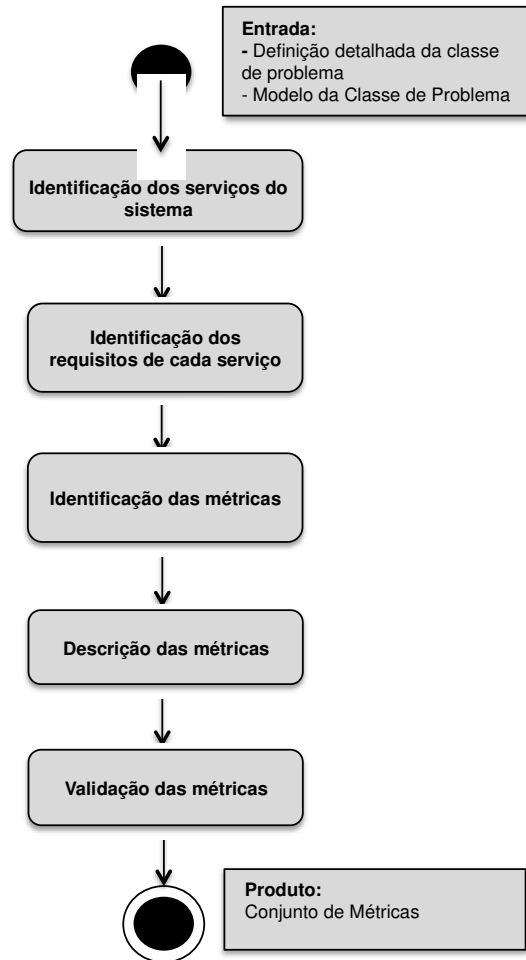


Figura 3.6: Definição das métricas.

3.6.5 Técnicas Sugeridas

Não consideramos ser necessária nenhuma técnica em particular para a execução dessa atividade visto que ela consiste basicamente em uma sequência de tarefas relativamente simples.

3.6.6 Produto

O produto desta fase é um conjunto de métricas. É importante que se documente também os serviços do sistema juntamente com seus respectivos requisitos a fim de que usuários do benchmark possam compreender a motivação por trás de cada métrica e possa estendê-lo a fim de atender novas demandas de pesquisa.

3.7 Escolha ou Desenvolvimento de um Simulador

3.7.1 Descrição

A última fase de PROBENCH diz respeito ao desenvolvimento ou escolha do simulador. Nela os desenvolvedores devem verificar se existe algum simulador que possa ser utilizado durante a realização dos experimentos. A existência e viabilidade de uso de um simulador é importante pelas seguintes razões:

- I. **Economia de recursos:** uma vez que não será necessário investir tempo e dinheiro no desenvolvimento de uma ferramenta, os desenvolvedores poderão acelerar o processo de criação do benchmark e investir o dinheiro em outras fases;
- II. **Base de usuários:** dependendo do simulador, os desenvolvedores do benchmark podem se beneficiar da base de usuários existente a fim de tornar o benchmark mais conhecido;

Essa fase é importante porque existem vários simuladores de propósito geral que podem ser utilizados para o estudo de SMA (Zoller et al., 2006). Sendo assim, é preciso que para cada simulador analisado, se verifique quais funcionalidades são atendidas (total ou parcialmente), potencial de incremento no conhecimento da área de pesquisa em particular, e requisitos do benchmark. Adicionalmente, essa atividade permite identificar as limitações em cada simulador existente.

Por outro lado, utilizar um simulador existente pode ter como desvantagem a falta de flexibilidade suficiente a fim de acomodar novas demandas ao benchmark e assim garantir sua **evolutibilidade**. Caso não seja identificado nenhum simulador que satisfaça os requisitos estabelecidos, será necessário então desenvolver um totalmente do zero.

3.7.2 Artefatos de Entrada

Os artefatos de entrada dessa fase são os seguintes: *objetivos do benchmark, definição detalhada da classe de problema e modelo da classe de problema.*

3.7.3 Papéis

Os atores que participam dessa fase do processo são o *pesquisador* e o *desenvolvedor de software*. O primeiro é responsável por definir quais aspectos do comportamento da

solução devem ser avaliados pelo benchmark e o segundo por adicionar outros aspectos que não foram previstos pelos pesquisadores.

3.7.4 Atividades

As atividades dessa fase conforme ilustrado na **Erro! Fonte de referência não encontrada.** são as seguintes: *identificação dos critérios de seleção do simulador, pesquisa dos simuladores existentes, aquisição do simuladores e implementação do simulador.*

Identificação dos critérios de seleção do simulador: nessa atividade os desenvolvedores devem definir quais são os critérios que devem orientá-los na busca pelo simulador a ser utilizado no benchmark. Alguns exemplos de critério são disponibilidade, possibilidade de alteração, portabilidade, desempenho, recursos disponíveis etc.

P.: *Quais são os critérios que devem ser utilizados na seleção do simulador?*

Pesquisa por simuladores que atendam aos critérios: tem por objetivo verificar se existem simuladores que atendam aos objetivos do benchmark sendo criado. Essa pesquisa deve ser realizada levando em considerações os critérios definidos na atividade anterior;

P.: *Existem simuladores que atendam aos critérios estabelecidos?*

Aquisição do simulador: caso tenha sido identificado um simulador que atenda aos requisitos do benchmark, deverá ser feita uma aquisição do mesmo;

Desenvolvimento do simulador: se não existir um simulador que possa ser utilizada na condução dos experimentos, será preciso desenvolver uma nova;

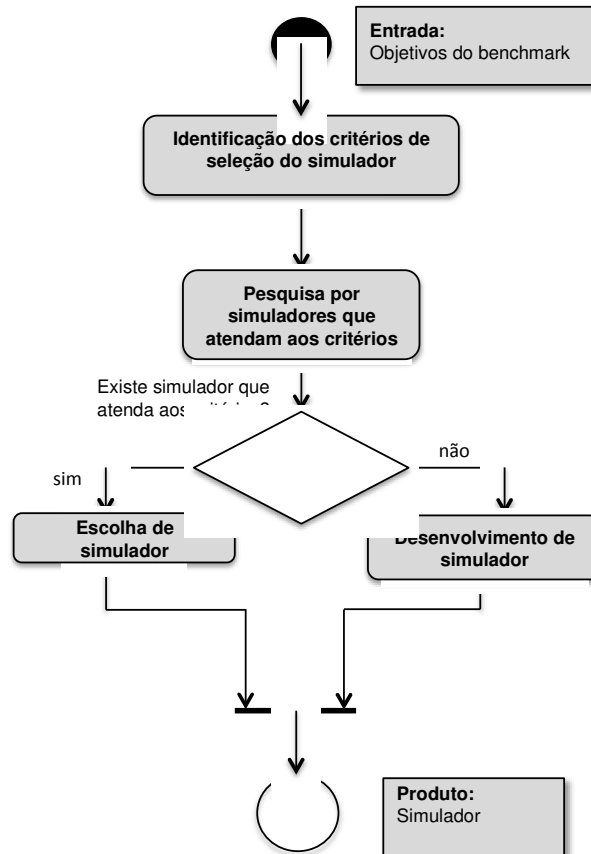


Figura 3.7: Desenvolvimento ou escolha do simulador.

3.7.5 Técnica Potencial

Caso seja necessário implementar um novo simulador, os desenvolvedores poderão empregar técnicas da Engenharia de Software, tais como Modelo em Cascata, Desenvolvimento baseado em protótipo, *Modelo Espiral*, *Processo Unificado*, *eXtremme Programming (XP)*, *Agile*, *Scrum*, *Crystal*, *Lean* etc. (Pressman, 2010).

3.7.6 Produto

O produto dessa fase deve ser um simulador selecionado pelos pesquisadores e/ou especialistas ou implementado pelo grupo envolvido com a criação do benchmark.

3.8 Conclusões sobre o Capítulo

Neste capítulo apresentamos o desenvolvimento de um processo *ad hoc* para auxiliar a criação de benchmarks para SMA. Ele consiste em um conjunto de atividades, onde

cada uma é responsável por produzir um ou mais artefatos necessário à realização de benchmarkings.

Conforme discutimos no capítulo 2, a principal motivação para o desenvolvimento de um processo para criação de benchmarks se deve à importância destes para a metodologia científica na realização de estudos comparativos em IA, mais especificamente em SMA.

Com o crescente interesse e adoção de soluções baseadas em IA, faz-se necessário ampliar, amadurecer e consolidar a adoção de benchmarks a fim de torná-la uma disciplina científica. Para isso faz-se necessária a existência de um processo sistemático e bem definido que possibilite que novos benchmarks sejam desenvolvidos de forma controlada e padronizada. Nesse sentido ressaltamos três vantagens no PROBENCH:

- (i) Simplificação do desenvolvimento de benchmarks ao guiar desenvolvedores através de um conjunto de fases bem definidas;
- (ii) Unificação da terminologia empregada no desenvolvimento de benchmarks (a ser apresentada no próximo capítulo);
- (iii) Padronização dos artefatos fundamentais de um benchmark.

Para chegarmos ao conjunto de atividades do PROBENCH foi realizado um estudo de diversos benchmarks tanto de SMA como de outros campos de IA o que nos permitiu verificar que benchmarks tem sido desenvolvidos informalmente, o que pode comprometer sua confiabilidade como experimental. Adicionalmente não há um compartilhamento detalhado e/ou completo das informações necessárias para se replicar experimentos.

Cada uma das fases será discutida em detalhes nos próximos capítulos. Começaremos descrevendo a atividade de Planejamento que consiste, dentre outras coisas, na definição da classe de problema.

Capítulo 4

Identificação da Classe de Problema

A primeira fase de PROBENCH é a *identificação da classe de problema*. Ela consiste em conduzir os desenvolvedores na busca por uma classe de problema que possa ser utilizada no benchmark.

Essa fase é a mais importante uma vez que a classe de problema identificada deve permitir que seu estudo e soluções propiciem uma maior compreensão dos fenômenos da área de conhecimento à qual ela se relaciona. Por outro lado, a classe de problema deve conter desafios que sejam o mais próximo possível daqueles encontrados no mundo real, tal que as soluções encontradas possam ser úteis do ponto de vista prático.

Neste capítulo iremos mostrar como essa fase foi utilizada na busca de uma classe de problema para a realização de pesquisas em SMA. Partimos do pressuposto que antes dela foi realizado um estudo a fim de descrever a motivação por um novo benchmark juntamente com seus objetivos e requisitos.

4.1 Identificação dos Critérios de Seleção da Classe de Problema

A escolha de uma classe de problema que seja ao mesmo tempo abrangente do ponto de vista da exploração do maior número possível de técnicas, e útil em relação às aplicações do mundo real é um dos maiores desafios quando se trata da concepção de um benchmark. Conforme visto no capítulo 2, problemas tais como jogar futebol, auxiliar no resgate de vítimas em desastres e organizar um sistema de compra de venda, apesar de suas contribuições ao contexto de SMA, ainda apresentam uma série de limitações.

A fim de que possamos chegar a um benchmark que contribua com a pesquisa em SMA, a classe de problema deve ser escolhida tendo como base critérios claros e definidos de acordo com a necessidade dos usuários. Essa escolha foi realizada a partir de dois contextos diferentes. Inicialmente fizemos uso da experiência do nosso grupo de pesquisa em do CIn-UFPE ao longo de vários anos de pesquisa no estudo do problema

da patrulha (Almeida et al, 2004; Talita et al., 2010; Menezes et al., 2010) e identificamos os seguintes critérios como sendo fundamentais:

Foco na tarefa: O benchmark deve permitir que o pesquisador possa concentrar sua atenção e esforço no estudo do problema, e não em detalhes secundários tais como o aprendizado das regras do benchmark (p. ex., qual o ângulo de visão dos agentes em uma partida de futebol? Em que situações o agente pode vender passagens de avião acima do valor máximo estipulado?) ou a utilização de softwares de apoio. Sendo assim, todos os detalhes subjacentes à simulação, tais como cenários, métricas, comportamento básico etc., devem ser fáceis de aprender e não representar um obstáculo na realização de experimentos.

Útil e realista: O benchmark deve apresentar um problema cuja solução seja útil para o desenvolvimento de alguma tecnologia que atende a algum problema importante ou permitir compreender melhor alguma teoria. A utilidade deve ser analisada também em termos dos investimentos necessários para desenvolver suas soluções. O benchmark deve estar relacionado a algum problema relevante do mundo real, tanto do ponto de vista científico como técnico, e esse problema deve representar também um grande avanço para a área. Em outras palavras, ele não deve ser um “toy-problem”. No entanto, ele não deve impor desafios ou metas de longo alcance tais como aqueles propostos nos desafios tais como *Teste de Turing* (1950), *Xadrez* (Levinson et al, 1991; Hayes e Ford, 1995; Neyborn, 2000; Hsu, 2002) e *RoboCup Soccer* (Kitano et al., 1996; Asada et al., 1999; Stone, 2003; Kalyanakrishnan et al., 2009). O problema nesses casos é que a competição por trás do desafio retira o foco do avanço científico e o transfere para o excesso de esforços em torno da vitória (Drogoul et al., 2007). Isso leva ao surgimento de soluções baseadas em força bruta ou com “artimanhas”, o que em nada contribui para o desenvolvimento da área de pesquisa em questão (Bayer et al., 2004; Drogoul et al., 2007).

Problema abstrato: O benchmark deve dispor de um problema que seja o mais abstrato possível. Dessa forma, a aplicação de diferentes técnicas e a generalização de resultados serão facilitadas. Problemas muito específicos tendem a reduzir o apelo, sobretudo em relação à captação de novos investimentos e expansão da comunidade em torno do benchmark. A solução poderá ser desenvolvida em vários tipos de plataformas, domínios, tecnologias etc. Por outro lado quanto mais abstrato é o problema mais reusável tende a ser a solução, pois ela não dependerá de detalhes muitos específicos para funcionar. Isso significa que, por exemplo, um

algoritmo desenvolvido para rastreamento em terrenos seja utilizado em aplicações de segurança em redes de computadores.

Documentabilidade: para que o benchmark possa ser largamente utilizado em sua comunidade de pesquisa, é preciso que exista o suporte necessário do ponto de vista do seu aprendizado. Além da documentação relacionada com a classe de problema em si, é preciso que se disponibilize também documentos que permite a fácil utilização dos softwares necessários, sobretudo se for preciso realizar qualquer modificação nos mesmos.

Em seguida, através da análise da literatura relacionada a benchmarks em SMA compilamos um conjunto de requisitos que, de acordo com seus atores, devem ser atendidos:

Problema atraente (Drogoul et al, 2007; Bayer et al, 2004): Todo benchmark possui um problema (concreto ou abstrato) para o qual deve ser desenvolvida uma solução. Esta afirmação nos leva a dois aspectos importantes. O primeiro diz respeito aos benefícios associados com a classe de problema do benchmark. É preciso de antemão saber quais são os ganhos teóricos ou práticos obtidos com seu estudo e consequente solução. O segundo está relacionado com a capacidade de atrair esforços e investimentos por parte de pesquisadores e investidores. Portanto, o problema deve ser atraente tanto do ponto de vista científico, fornecendo desafios interessantes para o desenvolvimento de novas teorias, como do ponto de vista comercial, a fim de aperfeiçoar produtos e serviços, e social, através de aplicações que auxiliem na solução de problemas que comprometam a qualidade de vida das pessoas.

Fácil compreensão (Bayer et al, 2004): A definição do problema deve ser a mais simples possível por maior que seja sua complexidade. Isso ajuda, dentre outras coisas, a minimizar problemas de comunicação e reduzir a possibilidade de interpretações diferentes. De preferência, ele deve ser tão simples que possa ser compreendido pelo público não-especialista.

Métricas objetivas (Drogoul et al., 2007): Métricas devem ser interpretadas objetivamente. Isso significa dizer que os resultados de um experimento científico devem permitir deduções claras e imediatas acerca do comportamento da solução que não dependam da opinião de um avaliador. Isso é diferente de métricas do tipo “vencer ou perder” ou “passar ou falhar”, as quais não fornecem nenhuma informação válida sobre o desempenho da solução.

Métricas claras e simples (Bayer et al, 2004; Huppler, 1999): O benchmark deve prover métricas claras e fáceis de entender e de implementar. Isso facilitará a dedução de informações a partir dos comportamentos observados durante os experimentos. É preciso também que elas possam ser calculadas imediatamente, pois em alguns casos é importante que o pesquisador possa acompanhar em tempo real a produção dos resultados enquanto o experimento é executado.

Abrangência no estudo de SMA (Huppler, 2009): agentes devem apresentar habilidades tais como autonomia, comunicação, negociação e cooperação, o que faz com que a pesquisa seja inerentemente multidisciplinar, envolvendo áreas como Psicologia, Economia, Linguística, Lógica, Pesquisa Operacional etc. Portanto, é fundamental que a classe de problema de um benchmark possibilite a adoção de técnicas variadas durante o projeto e a experimentação dos agentes. Huppler (2009) chama a atenção para o fato de que um benchmark deve essencialmente possuir uma grande aplicabilidade e uma audiência alvo abrangente a fim de ser relevante.

Evolucionável (Duchateau et al., 2006; Huppler, 2009): de acordo com Duchateau e colegas (2006), um benchmark deve evoluir ao longo do tempo a fim de permitir que novas hipóteses sejam avaliadas. Isso significa que novas funcionalidades ou aperfeiçoamentos devem ser realizados sem necessidade de reorganização significativa do benchmark. No entanto, isso só é possível se seu desenvolvimento se der de forma sistemática e houver alguma documentação da maneira em que ele foi conduzido. Para Huppler (2009), esse requisito é fundamental para a longevidade de um benchmark. Segundo o mesmo, um benchmark deve propor desafios significativos para uma área de problema durante vários anos. Dessa forma, a escolha da classe de problemas deve buscar o seguinte equilíbrio: propor um desafio suficientemente complexo e útil que demande esforços e valha à pena ser investigado durante vários anos e que ao mesmo tempo seja realizável.

4.2 Pesquisa por Classes de Problemas

Uma alternativa que tem se mostrado bastante promissora no estudo de SMA é o *problema da patrulha* ou (*Multiagent Patrolling*), o qual consiste em um grupo de agentes percorrendo uma área restrita a fim de monitorar ou supervisionar todos os seus locais (Abate, 1996).

Patrulhamento pode ser utilizado em uma série de aplicações do mundo real que envolve vigilância, inspeção ou controle distribuído, tais como vigilância de fronteiras e

de aeroportos, identificação de atualização de páginas web, patrulha militar, resgate em áreas perigosas, jogos eletrônicos, gerenciamento de falhas em redes de computadores etc. Essa característica aliada à possibilidade de utilizá-la como cenário para estudos de SMA e robótica, tem despertado a atenção de suas respectivas comunidades para o seu potencial. Por exemplo, permite o estudo de comportamentos inteligentes tais como planejamento, negociação, coordenação, comunicação, cooperação, aprendizado e tomada de decisão.

Conforme citado na seção anterior, nosso grupo de pesquisa vem trabalhando com esse problema há vários anos, tendo produzido uma quantidade considerável de trabalhos (Almeida et al., 2003; Almeida et al., 2004; Menezes et al., 2006; Moreira et al., 2008; Sampaio et al., 2010). Essa experiência associada aos motivos supracitados, permitiu-nos considerá-lo como sendo uma maneira para promover o avanço do estado da arte da área, além de contribuir para o desenvolvimento de aplicações de missão crítica. Diante dessas considerações, resolvemos verificar sua viabilidade como classe de problema na criação de um benchmark para SMA.

Na figura é apresentada um gráfico mostrando a distribuição da quantidade de trabalhos desenvolvidos no contexto de PMA em cada ano durante o período de 2002 a 2012. Através dele é possível perceber que tem havido um crescimento da pesquisa em PMA, inclusive com grupos de trabalho em diversos países tais como Brasil, França, EUA, Espanha, Israel, Portugal e Inglaterra.

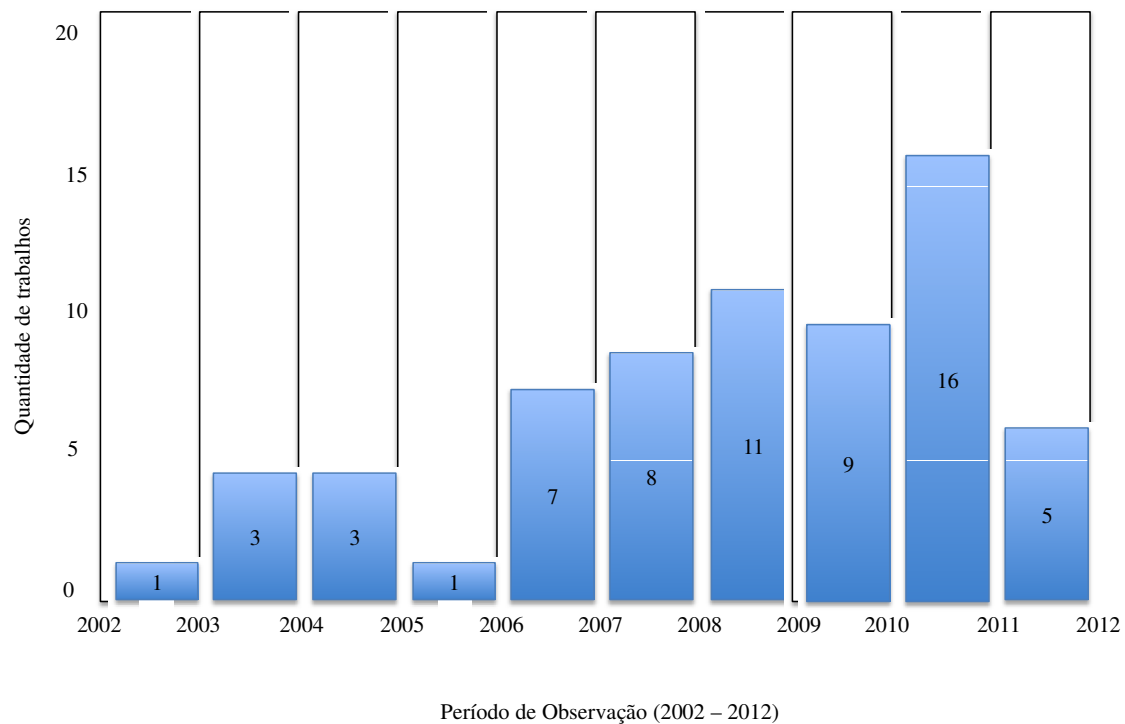


Figura 4.1: Quantidade de trabalhos em PMA no período de 2002 a 2012.

4.3 Descrição da Classe de Problema

Esta seção tem como objetivo descrever os elementos básicos do problema da patrulha: *agente patrulhador*, *região*, e *métricas*. A partir de suas características é possível elaborar diferentes variações do problema de forma a atender às necessidades dos pesquisadores.

4.3.1 Agente Patrulhador

Um agente patrulhador representa qualquer entidade física (soldado, bombeiro, agente de saúde, robô móvel) ou virtual (um personagem de um jogo eletrônico, um software monitorando uma rede de computadores ou páginas web recém-modificadas) cujo objetivo é patrulhar uma área.

A percepção consiste em informações recebidas do ambiente, tais como a *posição atual*, a *ociosidade dos vértices*, a *prioridade dos vértices*, *tamanho dos arcos do vértice corrente*, *instante de tempo atual* etc. A abrangência dos sensores pode variar desde sua vizinhança imediata até uma distância maior, dependendo do problema. Após a percepção, o agente deve escolher qual deve ser o próximo vértice a ser visitado,

observando qual proporciona uma maior recompensa ou um menor custo. Em seguida, ele se desloca até o vértice escolhido e um novo ciclo começa.

Apesar de que é possível realizar a patrulha com um único agente¹, o problema vem sendo tratado considerando a existência de múltiplos agentes. A vantagem de se adotar mais de um agente é que a complexidade da tarefa pode ser distribuída entre eles, reduzindo a área a ser coberta e melhorando o desempenho global da solução.

Os agentes podem trabalhar de duas formas: *isolada* ou *integrada*. No primeiro caso, a escolha pode ser devido à uma limitação do problema. Por exemplo, em regiões de guerra pode haver interceptação da comunicação entre os soldados e os inimigos identificarem o posicionamento de cada agente. Também pode ser por uma decisão de projeto, onde a área a ser patrulhada é dividida em várias partes e cada agente percorre apenas aquela a que lhe foi destinada.

No outro caso, os agentes se comunicam entre si como forma de suprir a falta de recursos ou pela impossibilidade de realizar a tarefa sozinho (Wooldridge, 2010; Jennings et al., 2003). Desta forma, os agentes podem se comunicar de forma direta ou indireta. Na comunicação direta os agentes trocam mensagens entre si, e na indireta, os agentes utilizam “sinais” (*flags*) espalhados no ambiente, imitando o mecanismo de comunicação das formigas (feromônios), ou por meio de uma estrutura compartilhada chamada de “quadro-negro” (*blackboard*), na qual os agentes escrevem e lêem as informações (Russel e Norvig, 2004).

4.3.2 Região

Região é uma área finita sobre a qual os agentes irão se deslocar a fim de realizar o patrulhamento, como por exemplo uma cidade, uma fábrica ou uma floresta.

Ela pode ser descrita de duas formas: *contínua* ou *discreta*. Na primeira forma a região é representada por um plano consistindo por infinitos pontos, de tal maneira que os agentes podem trafegar por todos eles a fim de alcançar os pontos de interesse. Na segunda forma, a região é representada por meio de uma abstração composta por um conjunto de vértices e arcos denominada de grafo. Os vértices representam os lugares que serão visitados, tais como residências, salas, servidores e agências bancárias, e os arcos representam os caminhos entre estes. Nesse caso, os agentes trafegam apenas entre os pontos de interesse.

¹ Chevalleyre (2004) estudou a utilização de um único agente para fins de patrulha em um grafo contendo dois vértices e a utilização de dois agentes patrulhando um grafo maior (6 vértices). No entanto, a abordagem utilizada (dividir e alocar o sub-grafo a um agente) remete à mesma abordagem utilizada no caso com um agente.

Em alguns contextos, o problema a ser abordado descreve naturalmente uma representação com essas características, como por exemplo, gerenciamento de redes de computadores, onde cada vértice representa um equipamento a ser monitorado e as ligações físicas entre eles são representadas por arcos. No entanto, existem contextos cujas regiões são inerentemente contínuas, tais como aplicações onde são utilizados robôs. Nesse caso, os agentes terão uma quantidade infinita de posições a analisar o que será inviável. A fim de simplificar esses casos, uma técnica chamada de *esqueletonização* é utilizada (Machado, 2002). Ela consiste em converter a imagem de um ambiente real em uma estrutura de dados abstrata discreta. Essa técnica consiste em transformar uma representação contínua ou espacial em uma representação discreta, por meio da identificação dos possíveis caminhos entre os vértices, conforme visto na figura 4.2. Esta transformação pode ser feita usando várias técnicas diferentes, tais como *diagramas de Voronoi*, *grafos de visibilidade*, *C-cells*, *métodos de contorno* e *traçado manual* (Machado, 2002).



Figura 4.2: Processo de Esqueletonização: (i) Mapa da Cidade; (ii) Pontos escolhidos para monitoração ligados pelos caminhos a serem percorridos pelo agente; (iii) grafo resultante.

Dado o grafo obtido a partir da área, a tarefa da patrulha passa a ser a de visitar todos os vértices (Santana et al., 2004). Formalmente um grafo G pode ser definido como uma tupla formada pela tupla $\langle V, E \rangle$, onde V é o conjunto de vértices e E é o conjunto de arcos.

Um grafo pode ter propriedades como, por exemplo, ser dinâmico: os arcos e os vértices aparecerem ou desaparecerem durante a patrulha. Esse tipo de situação descreve, por exemplo, no primeiro caso, cenários como patrulhamento em áreas urbanas onde uma rua pode ficar temporariamente interditada. Similarmente, uma agência bancária pode ficar fechada durante uma época ou um servidor web ficar

inativo por um período de tempo. Outra propriedade diz respeito à possibilidade dos arcos serem unidirecionais ou bidirecionais, representando, por exemplo, ruas nas quais o tráfego pode fluir em um ou dois sentidos.

Para a realização dos estudos do problema da patrulha, vários cenários têm sido propostos, tais como *Mapa A*, *Mapa B*, *Corredor*, *Circular*, *Ilhas* e *Grade* (Machado et al., 2002; Almeida, 2004) conforme apresentado na figura 4.3. Eles representam terrenos que apresentam características interessantes de serem estudados por causarem algum impacto no desempenho das arquiteturas usadas no desenvolvimento dos agentes.

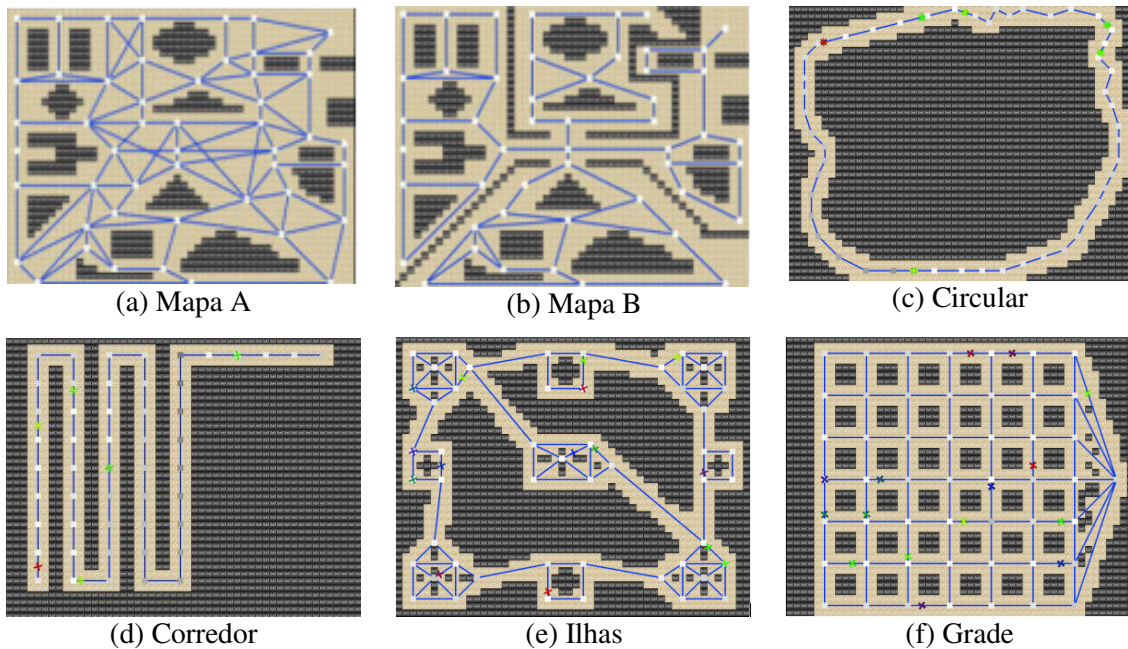


Figura 4.3: Exemplos de regiões com diferentes topologias (Almeida et al., 2004).

4.3.3 Métricas

Métricas são a forma de avaliar o desempenho de uma arquitetura SMA e assim poder permitir sua comparação com outras existentes. Sua definição depende do objetivo da patrulha (Lass et al., 2010).

Em grande parte dos trabalhos abordando o problema da patrulha o objetivo é minimizar o intervalo de tempo entre duas visitas consecutivas a um mesmo vértice. Conforme será melhor explicado na próxima seção, isso decorre da falta de

conhecimento que os agentes têm do mundo. A esse intervalo é dado o nome de *ociosidade do vértice*. A partir dele várias métricas foram definidas: *ociosidade instantânea do vértice*, *ociosidade média*, *ociosidade máxima*, *ociosidade relativa instantânea* etc. (Almeida et al., 2003; Almeida et al., 2004; Sempé e Drogoul, 2003; Chevaleyre, 2004; Chu et al., 2006; Sampaio et al., 2010).

Alguns trabalhos propuseram novas métricas baseadas na *frequência de visitas*, as quais consideram basicamente a quantidade de visitas à cada vértice do grafo ao longo da simulação. Elmaliach e colegas (2006) propuseram métricas baseadas nesse conceito tais como *frequência média por vértice*, *variância das frequências por vértice*, e a *frequência mínima por vértice*.

O trabalho de Sampaio e colegas (2010) aperfeiçoou o entendimento sobre ociosidade. A compreensão até então existente era que a cada vez que um vértice recebe uma visita, o valor de sua ociosidade é zerado, o que impedia que se conhecesse a distribuição das visitas. A definição de métricas é definida por meio do intervalo médio, intervalo máximo, e desvio padrão dos intervalos e média quadrática ponderada dos intervalos entre visitas. Caso a região possua vértices com prioridades diferentes, é preciso adaptar a métrica para incluir essa informação (Araújo, 2010). Nesse caso pode ser utilizada a média quadrática ponderada dos intervalos, onde as prioridades dos vértices serão representadas pelos pesos dos intervalos.

Poucos trabalhos apresentam sugestões de métricas com respeito à detecção de eventos (Ruan et al, 2005; Ahmadi e Stone, 2006; Agmon, 2010). Em todos os casos é utilizada a função de utilidade dos agentes (*reward*) como critério de avaliação do desempenho dos agentes.

4.4 Análise dos Critérios de Escolha

Apesar dessa constatações iniciais, faz-se necessário analisar se a classe de problema atende aos critérios elencados na seção anterior para assim podermos confirmar sua utilização no benchmark para estudo de SMA. Duas questões merecem ser ressaltadas nessa atividade: a primeira é importante que essa análise seja feita em conjunto pelos desenvolvedores do benchmark uma vez que os critérios são subjetivos, e a segunda é que dificilmente uma classe de problema vai atender a todos os critérios, devendo os desenvolvedores analisar se aqueles não atendidos podem ser relevados, se a classe de problema deve ser modificada a fim de atendê-lo ou se é melhor pesquisar por outra classe de problema.

- **Fácil compreensão:** conforme visto na seção, PMA pode ser compreendida por meio de uma definição informal retirada de um dicionário. Além disso, seu comportamento pode ser descrito por meio de

poucas entidades (agente, grafo e métricas) e por meio de um conjunto bastante resumido de ações (mover, parar, escolher vértice).

- **Foco na tarefa:** consideramos que PMA atende a esse requisito, pois ela consiste de um problema simples de entender e de utilizar. Dessa forma, complexidade extra não é adicionada à realização dos experimentos.
- **Útil e realista:** uma forma de constatar se PMA atende a esse critério é observando que a tarefa de patrulhar é bastante comum em uma série de aplicações importantes do dia a dia, como por exemplo, vigilância de fronteira, supervisão de aeroportos, gerenciamento de redes de computadores, resgate de desaparecidos etc.
- **Problema abstrato:** sua flexibilidade em permitir que uma grande variedade de aplicações seja desenvolvida por meio dela como, por exemplo, vigilância de fronteira, supervisão de aeroportos, gerenciamento de redes de computadores, resgate de desaparecidos etc.
- **Abrangência no estudo de SMA:** PMA permite a exploração de técnicas relacionadas com comunicação, negociação, cooperação, coordenação, autonomia de decisão, aprendizado multiagente etc.
- **Problema atraente:** a patrulha pode ser considerada uma classe de problema atraente por representar uma demanda de grande relevância no mundo real, como, por exemplo, monitoração, vigilância, gestão de desastres etc. Adicionalmente, por meio dessa classe de problema é possível realizar uma grande quantidade de estudos tanto em SMA como em robótica.
- **Métricas objetivas:** a avaliação de desempenho dos agentes em PMA é dada por métricas tais como ociosidade instantânea, ociosidade média e maior ociosidade instantânea, as quais são definidas por meio da contagem de tempo. Dessa forma, podemos afirmar que suas métricas não dependem da interpretação subjetiva dos usuários.
- **Métricas claras e simples:** de acordo com os trabalhos realizados em PMA, as métricas utilizadas são bastante claras e simples, facilitando tanto o entendimento como sua implementação no simulador.
- **Evolucionável:** consideramos que PMA atende a esse requisito pelo fato de que é possível estendê-lo a fim de atender novas demandas. É possível constatar essa afirmação pelo fato de que nos primeiros trabalhos em PMA (Machado et al., 2003, Almeida et. al., 2004) se propunha apenas agentes visitando os vértices e tentando minimizar a ociosidade de visitas, enquanto trabalhos mais recentes trazem a idéia de agentes invasores que buscam chegar a determinados vértices ou entrar no grafo

(Ahmadi e Stone, 2005; Agmon et al., 2008; Basilico et al, 2009a; Basilico et al, 2009b; Basilico et al, 2009c).

4.5 Conclusões sobre o Capítulo

O objetivo desse capítulo foi apresentar a fase de *seleção da classe de problema* do benchmark de acordo com o conjunto de diretrizes apresentado no capítulo 3. Essa fase leva em consideração a necessidade de lidarmos com a complexidade de se criar um benchmark. A idéia é que o pesquisador possa dispor de um “*mapa*” que o conduza ao longo de todo o processo.

Através dele identificamos na literatura de SMA uma classe de problema para a qual existe uma demanda importante tanto do ponto de vista teórico como prático para a criação de um benchmark. Esse problema é denominado de patrulha multiagente () e consiste em um grupo de agentes percorrendo uma região em intervalos regulares.

pode ser utilizado em uma série de aplicações do mundo real que envolve vigilância, inspeção ou controle distribuído, tais como vigilância de fronteiras e de aeroportos, identificação de atualização de páginas web, patrulha militar, resgate em áreas perigosas, jogos eletrônicos, gerenciamento de falhas em redes de computadores etc. Ela também permite o estudo de comportamentos inteligentes importantes, tais como planejamento, negociação, coordenação, comunicação, cooperação, aprendizado e tomada de decisão. Dessa forma, ela tem despertado a atenção das comunidades de SMA e robótica.

Entretanto, a forma a pesquisa em torno dessa classe de problema vem sendo realizada é confusa e desorganizada. Através de uma investigação da literatura, identificamos que não existe um consenso acerca do que realmente define como classe de problema. Em seguida, apresenta restrições quanto à comparabilidade de resultados o que impede sua adoção de forma abrangente. Por último, vários trabalhos tem se concentrado em torno de cenários bastante simplificados, o que impede sua larga adoção do ponto de vista prático.

Através deste capítulo pudemos apresentar as seguintes contribuições: desenvolvimento de um conjunto de diretrizes com o objetivo de informar os pesquisadores quais atividades devem ser realizadas a fim de orientar as demais etapas. A segunda foi a organização e classificação dos trabalhos realizados de acordo com o objetivo dos agentes, identificação e caracterização de um conjunto de limitações na forma como vem sendo estudada. A segunda foi uma análise minuciosa dos problemas na pesquisa corrente de PMA. Através do entendimento obtido a partir dessa análise poderemos aperfeiçoá-la e estendê-la no sentido de criarmos um benchmark para SMA.

No próximo capítulo iremos descrever a fase de modelagem da classe de problema.

Capítulo 5

Detalhamento da Classe de Problema

Uma vez que já realizamos a *seleção da classe do problema do benchmark*, podemos iniciar a fase seguinte do PROBENCH, que é o *detalhamento da classe de problema*. Seu objetivo principal é fornecer um entendimento mais aprofundado da classe de problema que será utilizada no benchmark.

Inicialmente iremos realizar uma organização conceitual a fim de superar as limitações identificadas na seção 5.1. Essa organização será feita por meio de uma taxonomia e uma nova definição de .

Em seguida, iremos apresentar uma classe de problema denominada de PMAOE (*Event Oriented MultiAgent Patrolling*), na qual os agentes tem conhecimento explícito acerca dos eventos que ocorrem no mundo. Para isso apresentaremos uma discussão aprofundada a respeito de eventos sob a luz da Metafísica e, em seguida, iremos propor um conceito do termo levando em consideração as particularidades do problema da patrulha. Também apresentaremos uma taxonomia de eventos a fim de definirmos precisamente os elementos que compõem o termo e criar uma estrutura conceitual que possa ser usada nos experimentos, conforme será explicado no próximo capítulo.

5.1 Identificação de Limitações

Ao longo da análise da literatura de , constatamos a existência de diversas limitações as quais prejudicam sua adoção como uma classe de problema para o desenvolvimento de um benchmark para SMA. As limitações podem ser classificadas em três categorias principais: *falta de um entendimento claro e consensual de* ; *restrições de comparabilidade*, e *cenários super-simplificados*.

Nas próximas subseções descreveremos essas limitações com o propósito de fornecer soluções e contribuir para a definição de como um benchmark.

5.1.1 Falta de Entendimento Claro e Consensual de

Um dos grandes problemas na pesquisa em é a falta de um entendimento claro e consensual do que é o problema da patrulha. Isso ocorre devido ao fato que as pesquisas realizadas utilizam como base um conceito informal do termo “patrulha”, retirado de um dicionário (Abate, 1996). Nos próximos parágrafos iremos discutir quais são essas limitações.

Diferenças entre Definição do Problema e da Solução: Esse problema está relacionado com a falta de uma definição clara, sem ambiguidades e, mais importante, de um consenso em torno de uma única definição de . Conforme supracitado, a definição mais utilizada nos trabalhos existentes vêm de uma definição informal, retirada de um dicionário (Abate; 1996). Ela mistura o objetivo (proteção, supervisão, segurança etc.) com o método de alcançá-lo (percorrer uma área em intervalos regulares). Uma prova dessa afirmação é a variedade de trabalhos onde o método de patrulhamento é usado como definição. Dentre os trabalhos disponíveis na literatura podemos destacar a patrulha baseada em frequência (Elmaliach et al., 2006), patrulha adversarial (Amigoni et al., 2009), patrulha baseada em ociosidade (Elor e Bruckestein, 2009; Ahmadi e Stone, 2005; Almeida et al., 2003), e orientada a eventos (Ahmadi e Stone, 2005). Apesar de sua indiscutível utilidade para o surgimento e desenvolvimento da área, faz-se necessário buscar uma definição adequada do problema.

Falta de uma Caracterização de : Apesar de que na literatura sobre , alguns trabalhos compartilham certas características, tais como representação do terreno por meio de grafo, região representada através de grafos, métricas baseadas em ociosidade etc., existem inúmeras diferenças quanto à maneira como os trabalhos caracterizam . Por exemplo, os grafos utilizados em determinados trabalhos são geralmente de tamanho e topologias diferentes de outros; alguns autores consideram a existência de agentes adversários tentando invadir o grafo (ou determinados vértices), enquanto outros não consideram nenhum adversário; alguns têm agentes com percepção limitada enquanto que em outros, varia; em alguns casos a sociedade é aberta, enquanto em outras são fechadas. Há também agentes com percepção perfeita e imperfeita (Agmon, 2004). Mesmo as métricas utilizadas variam bastante. Enquanto alguns autores focam na ociosidade como métrica fundamental, outros preferem a recompensa obtida pelos agentes, e há também aqueles que avaliam seus agentes observando o custo mínimo total.

Falta de uma Terminologia Bem Definida e Comum: Uma questão fundamental na consolidação de um domínio do conhecimento é a definição de uma terminologia que represente precisamente seus constituintes em relação à sua realidade e que possua uma formalidade capaz de restringir sua interpretação à sua real significância. Terminologia é o recurso indispensável para representar e comunicar o conhecimento especializado. Para isso, os termos devem refletir a estruturação conceitual desse domínio e permitir a comunicação especializada por meio da ordenação do pensamento e da transferência eficaz e sistematizada do conhecimento. Nesse sentido, podemos afirmar que um dos problemas da diz respeito à falta de uma terminologia bem definida e comum. Isso significa dizer que além de não haver uma escolha adequada e/ou estabelecida precisamente dos termos, não há um consenso acerca tanto da definição do próprio problema do patrulhamento como de quais são os termos fundamentais e seus significados dentro desse contexto. Como consequência dessa falta de atenção para com esse aspecto do domínio do problema da patrulha, podemos elencar os seguintes problemas:

- Prejudica o entendimento do problema e, conseqüentemente, a criação de uma definição comum;
- Prejudica a comunicação tanto intra como intergrupos de pesquisas, reduzindo as chances de troca e reuso de experiência;
- Reduz a possibilidade de comparações.

Falta de um Modelo de Referência: Esse problema é uma consequência do descrito anteriormente. O fato é que como não há um entendimento comum do que é , cada iniciativa desenvolve sua interpretação quase particular do problema da patrulha. Assim, as intersecções entre os trabalhos não estão claras (ou sequer visíveis) e elas não englobam todos os trabalhos. Em outras palavras: existem vários entendimentos do que uma instância do problema deve abranger. Muitas vezes os trabalhos deixam de apresentar informações a respeito de elementos do problema ou minimizam sua importância. É fundamental, sobretudo para fins de comparação, que um conjunto de elementos, mesmo que mínimo, seja considerado pelos pesquisadores. Então questões como essas são importantes para padronizarmos os problemas:

- Como o terreno vai ser representado?
- Quais os tipos e tamanhos dos grafos adotados?
- A sociedade é aberta ou fechada?

- O que o agente está buscando durante a patrulha?
- O que o agente consegue perceber do ambiente?
- Qual o alcance da percepção dos agentes?

A idéia, portanto, é que esse modelo possa servir de referência na definição de uma instância da .

5.1.2 Restrições na Comparabilidade

Comparabilidade é uma propriedade essencial para qualquer investigação científica de cunho empírico. Ela permite identificar os melhores resultados dentre um conjunto de alternativas possíveis e assim avaliar o progresso científico de uma determinada área do conhecimento (Hanks et al., 1993). No entanto, no que diz respeito ao problema da patrulha, identificamos várias restrições que limitam a comparação entre os trabalhos.

Falta de Métricas Padronizadas: Existem vários critérios de desempenho e, conseqüentemente, diversas métricas nos trabalhos existentes, tal como discutido em 4.3.3. Por exemplo, ociosidade, frequência, adversários localizados, intervalo médio de detecção de eventos etc. Duas questões se destacam nesse problema: a primeira é que métricas se confundem com estratégia da solução. Isso decorre do fato que a forma de avaliar o trabalho está intrinsecamente ligada à forma como a estratégia da solução para o problema foi projetada, o que no nosso entendimento, cria um viés metodológico; a segunda é que essa diversidade de métricas praticamente inviabiliza qualquer tentativa de comparar o desempenho das soluções, visto que elas não são compartilhadas.

Falta de uma Definição para o Termo Evento: Apenas recentemente é que trabalhos lidando diretamente com eventos surgiram em (Mazda e Stone, 2006; Agmon et al., 2009; Agmon, 2010; Agmon et al., 2011). No entanto, nenhum deles oferece uma definição ou uma caracterização para o termo. Dessa forma, seu uso nos referidos trabalhos é totalmente intuitivo e assumido a priori. Para que possamos explorar todo o potencial que acreditamos ter essa abordagem, faz-se necessário, antes de qualquer coisa, realizarmos um estudo aprofundado e assim provermos uma definição útil e aos nossos objetivos, e embasada na Filosofia, sobretudo quantos aos aspectos metafísicos. Outro problema observado nessas abordagens é que, por não haver uma definição do que é um evento, não se leva em consideração aspectos fundamentais como causalidade, individuação, duração etc. (Bennet, 1989; Hoefler, 2010; Casati e Varzi, 2010; Falcon, 2011). Acreditamos que adotar esse conceito sem a devida compreensão pode levar a

resultados que não podem ser extrapolados para a prática do dia-a-dia, restringindo-o totalmente a problemas excessivamente simples e rudimentares.

5.1.3 Cenários Super-simplificados

Apesar das contribuições dadas à e, consequentemente, a SMA, pudemos observar que, sob uma perspectiva prática, a grande maioria é bastante limitada, o que inviabiliza induzir seus resultados para problemas do real. Apesar de que essa constatação não os inviabilizam quanto às contribuições científicas, a adoção de cenários super-simplificados reduz o poder explicativo de fenômenos do mundo real, restringindo-os a exemplos meramente teóricos. Sendo assim, destacamos dois aspectos fundamentais que limitam a adoção de relacionados ao uso de cenários super-simplificados.

Grafos rudimentares: Apesar de que alguns trabalhos consideram a existência de grafos bastante interessantes do ponto de vista da experimentação empírica (Almeida et al, 2002), outros lidam com grafos bastante simples. Por exemplo, Chevalleyre (2004) aborda uma região circular enquanto Agmon e colegas (2008a e 2008b) lidam com o patrulhamento sobre linhas. Apesar da utilidade desse tipo de grafo em alguns domínios (p. ex. patrulhamento de fronteiras), o algoritmo desenvolvido para esses casos provavelmente não terá o mesmo desempenho se for aplicado em grafos irregulares tais como Mapa A ou Mapa B (Machado et al., 2002).

Ausência completa de conhecimento do mundo: Nos trabalhos de , principalmente MATP, os agentes patrulhadores desconhecem quase que completamente o que está sendo procurado e/ou sua natureza, ou seja, as características que descrevem o objeto ou fenômeno de interesse. Essas características podem ser o comportamento, causalidade, tipo de localização etc. Dessa forma, os agentes baseiam suas decisões apenas a partir da sua posição e da ociosidade dos vértices. Isso nos permite fazer um paralelo entre com os algoritmos de busca cega e os algoritmos de busca informada. Enquanto que no primeiro caso os agentes procuram atingir um estado de mundo desejável a partir de heurísticas de tentativa e erro, no outro os agentes realizam a tarefa a partir de símbolos que descrevem o mundo, o que permite um processo de inferência mais adequado (Schildt e Norvig, 2004). Levando em consideração a existência de cenários nos quais as características do objeto de interesse possam ser descritas e conhecidas pelos agentes, teremos patrulhas semelhantes àquelas que encontramos no mundo real (exército, salva-vidas, bombeiros, vigilância etc.). Dito isso, é possível projetar agentes inteligentes o suficiente para evitar que determinados

eventos venham a causar grandes prejuízos ou a perda de vidas na medida em que, de posse de um conhecimento prévio sobre os eventos, os agentes possam evitá-los ou minimizar os danos de sua ocorrência. Na forma em que se encontra o estado da arte da pesquisa em , sem o devido conhecimento do que é um evento, esse tipo de patrulha não é possível.

5.2 Aperfeiçoamento da Classe de Problema

5.2.1 Uma Nova Definição para Patrulha Multiagente

Conforme discutido na subseção 5.1.1, existem diferentes entendimentos acerca do que é o problema patrulha () o que torna sua compreensão uma tarefa confusa. Com o objetivo de realizar uma organização conceitual de , consideramos que é de grande importância a formulação de uma definição que descreva de forma precisa o que é uma patrulha nesse contexto, sem entrar em detalhes a respeito de sua solução. Além disso, a definição deve levar em consideração aquelas características que consideramos como essenciais na sua realização. Por fim, ela deve abranger todos os tipos de patrulhas.

Definição 1 (Patrulha): *É o problema que consiste em um grupo de agentes percorrer continuamente uma região durante um intervalo de tempo finito com o objetivo de identificar a ocorrência de eventos, tendo eles conhecimento ou não de sua natureza. Os agentes devem selecionar e visitar os lugares da região de forma a atender um ou mais critérios de desempenho.*

De acordo com essa definição, a região patrulhada deve ter tamanho limitado, do contrário, sempre haveria uma quantidade infinita a serem visitados, o que inviabiliza o propósito da tarefa, tais como ocorre em problemas de exploração.

Similarmente, para fins de estudo, a patrulha deve ter uma duração limitada a fim de que possamos utilizá-lo como método de experimentação em SMA. Os agentes tem como objetivo a identificação de eventos. No entanto, conforme discutiremos em detalhes na seção 5.3.2, em PMAT e PMAA os agentes desconhecem o que estão tentando identificar, pois não possuem um conhecimento explícito acerca dos eventos. Dessa forma, a estratégia adotada é a redução dos intervalos ou aumento da frequência de visitas ao mesmo vértice, aumentando as chances de sucesso. É importante ressaltar que, apesar do objetivo mencionado (*detectar a ocorrência de eventos*), em grande parte dos trabalhos não existe a preocupação em se verificar se as visitas conseguem realmente atendê-lo.

Os agentes devem escolher os lugares que serão visitados (ou *pontos de interesse*) levando em consideração algum critério de desempenho estabelecido pelos pesquisadores. Esses critérios representam a capacidade dos agentes realizarem de forma satisfatória a tarefa, seja maximizando ou minimizando alguma medida. Por exemplo, no trabalho de Portugal e Rocha (2010) é utilizada a maximização da frequência média de visitas, enquanto que no trabalho de Almeida e colegas (2003) procura se minimizar a ociosidade média. O critério de desempenho é também chamado de critério-objetivo a fim de representar a idéia de que os agentes são projetados com o intuito de atenderem à alguma expectativa de sucesso.

5.2.2 Nossa Proposta: Patrulha Multiagente Orientada a Eventos

Conforme visto no capítulo anterior, existem vários problemas que impedem a adoção de como um benchmark para SMA. A fim de superá-los, propomos uma abordagem existente na literatura denominada PMAOE (*Event Oriented Multiagent Patrolling*), na qual os agentes podem tomar decisões levando em consideração o conhecimento sobre os eventos que podem ocorrer. Diferente dos trabalhos existentes (Mazda e Stone, 2006; Agmon et al., 2009; Agmon, 2010), em PMAOE os eventos são descritos de forma a considerar suas características fundamentais, tais como causalidade, duração, localização, impacto etc.

Além disso, esses trabalhos se limitam a grafos bastante rudimentares, o que impede sua adoção de forma abrangente do estudo de técnicas de SMA. Sendo assim, propomos PMAOE com o objetivo de que ele possa contemplar as necessidades da comunidade em torno de um benchmark que permita explorar as diversas técnicas de SMA no desenvolvimento de soluções para problemas realistas.

Na próxima seção apresentaremos uma descrição detalhada do conceito de evento levando em consideração a fundamentação existente em Metafísica, que é o ramo da Filosofia encarregado de estudar as chamadas “primeiras coisas” (Aristóteles, 2010).

5.2.3 Eventos de Acordo com a Metafísica

Eventos são o que de fato movem o mundo: a queda da bolsa de Nova Iorque, uma criança chorando, o gotejar de uma torneira, o assassinato de Júlio César por Brutus, o sorriso de uma pessoa, a reação eletroquímica de uma molécula, uma pessoa mastigando, uma festa, o piscar de olhos, a colisão de um automóvel, o acender de um isqueiro, a explosão de uma bomba nuclear, uma corrida de cavalos, o nascimento e a morte de um ser vivo. Mas o que faz acontecimentos tão distintos como o assassinato de Julio Cesar e a queda da bolsa de Nova Iorque pertencerem a uma mesma categoria

metafísica? Quais propriedades permitem que eles possam ser estudados sob uma mesma perspectiva? Por que é importante responder a essas perguntas?

Para responder a essas perguntas e prover um esquema teórico que seja útil para a criação do benchmark proposto, precisamos entender o conceito de eventos de forma detalhada e coerente com as investigações realizadas no campo da metafísica.

De acordo com Bennett (1988), um evento é qualquer coisa que acontece. Esse conceito está intimamente relacionado à noção de ação, da realização de algum acontecimento capaz de mudar o estado de alguma coisa no mundo. Desta forma, podemos afirmar que eventos acontecem a todo instante e em todos os lugares, não importa se são perceptíveis ou não.

Metafísica é o ramo da Filosofia que estuda a natureza fundamental do ser e do mundo. De acordo com Garret (2006), Metafísica se dedica à investigação das características mais gerais e onipresentes da realidade, tais como existência, espaço, tempo, causação, tempo, objeto, propriedade etc., e de tópicos específicos tais como livre arbítrio e identidade pessoal. Dentre esses tópicos, eventos desempenham um papel fundamental na compreensão de inúmeros fenômenos. Em função disso, *teorias de eventos* surgiram a partir do desafio de se definir adequadamente o que é um evento, quais suas características e aplicabilidade.

Uma teoria de eventos é importante por várias razões (Davidson, 1980; Kim, 1993; Brand, 1997; Scheneider, 2010; Casati e Varzi, 2010). Primeiro, ela nos permite descrever uma condição de existência para os eventos. Isso ajuda a entender melhor como esse conceito difere de outras categorias metafísicas. Segundo, ela propõe uma condição de identidade, ou seja, sob quais condições dois eventos são iguais. Em outras palavras isso significa dizer em que condições podemos afirmar que um evento é único. Terceiro, através de uma teoria de eventos podemos definir sob quais condições um determinado evento pode ocorrer e quais suas consequências.

Nesse sentido, diversas teorias surgiram, cada uma motivada por uma interpretação diferente (Kim, 1993; Brand, 1997; Davidson, 1980). Nos próximos parágrafos discutiremos as principais características das teorias apresentadas.

A teoria de Kim (1993) é chamada de *Exemplificação-Propriedade*. De acordo com a mesma, eventos são representados por meio de uma tupla formada pelos elementos $\langle x, P, t \rangle$, onde x representa um ou mais objetos envolvidos no evento, P as propriedades desses objetos ou uma relação entre eles. Por último, o tempo t é o instante corrente em que o evento acontece. Um conjunto de propriedades indica a condição atual do objeto, geralmente transmitindo um sentido de ação. De acordo com essa teoria um evento $\langle x, P, t \rangle$, existe se e somente se o objeto x possui P no tempo t . Isso significa que para x participe de um evento no tempo t ele tem que exemplificar a propriedade P

naquele instante. Isso significa que a propriedade de “João” é que ele passeou à noite, i.e., o evento será $e = \langle \text{"João"}, \text{"passeou"}, \text{"noite"} \rangle$.

Uma propriedade dessa definição é que um evento $\langle x, P, t \rangle = \langle y, Q, u \rangle$ se e somente $x = y$, $P = Q$ e $t = u$. Isso significa que a única possibilidade de dois eventos serem iguais é se os objetos forem os mesmos (x e y), com a mesma propriedade (P e Q) e no mesmo instante (t e u). Ou seja, cada instância de evento é único. Esse princípio é às vezes chamado de “não-duplicação”. Adicionalmente, eventos têm uma locação espaço-temporal, são irrepetíveis e incluem não apenas mudanças, mas também estados e condições.

Myles Brand (1997) critica a idéia de que eventos devem ter necessariamente um objeto, ou seja, um *sujeito da ação*. Ele cita os eventos naturais, tais como mudanças climáticas (chover, nevar etc.), mudanças de iluminação (escurecer, clarear etc.) e alterações em um campo magnético, como exemplos de eventos onde não existem objetos desempenhando o papel de sujeito (*constituente*), tais como apresentados por Kim. Ao invés do objeto, Brand sugere que o *espaço* seja o constituinte do evento (Schneider, 2010). Outra crítica diz respeito às propriedades. Primeiro, ele questiona sobre a falta de informações na teoria de Kim sobre quais tipos de propriedades podem consideradas como propriedades constitutivas. Segundo, ele critica o critério de igualdade (Brand, 1997) utilizado por Kim, pois, segundo ele, não seria possível determinar quando dois eventos têm as mesmas propriedades.

A teoria desenvolvida por Davidson se preocupa com a questão da igualdade dos eventos. Ela pode ser dividida em duas partes de acordo com o critério empregado: (i) *critério causal* e (ii) *critério espaço-temporal* (Davidson, 1980; Scheneider, 2010; Casati e Varzi, 2010). No primeiro caso, afirma-se que dois eventos são iguais se e somente se tem as mesmas causas e efeitos. Na sua argumentação, Davidson assume que eventos são ocorrências particulares não-repetíveis. No entanto, uma crítica feita à essa definição é quanto a sua *circularidade*. Dois eventos e e e' são iguais se eles tem as mesmas causas e consequências, as quais chamaremos de c o conjunto de causas de e e de c' o conjunto de causas de e' . Naturalmente que c e c' são eventos e se estamos em busca de afirmar que e e e' tem a mesma causa, estamos tentando provar que $c = c'$. Todavia, para provar esse argumento, c e c' teriam que ter as mesmas causas e consequências, o que levará indefinidamente à busca de uma causa final. O mesmo argumento vale para as consequências do evento.

Dadas essas teorias, podemos construir uma definição de evento para o problema da patrulha coerente com a Metafísica. Isso será feito ao longo das próximas seções. No entanto, antes de apresentá-la, discutiremos alguns exemplos a fim de ilustrar como o conceito de evento pode fornecer cenários interessantes para o estudo da patrulha.

5.2.4 Exemplo Ilustrativo

Considere que o grafo apresentado na figura 5.1 representa uma cidade e os vértices representam diferentes tipos de lugares de uma cidade. Um grupo de agentes tem como objetivo visitar os vértices a fim de verificar se algum incidente está ocorrendo e consequentemente assegurar a segurança de cada um dos locais.

Suponha que um agente desse grupo se encontra inicialmente no vértice v_1 e que o vértice v_2 representa uma praia onde podem ocorrer afogamentos, o vértice v_3 representa uma fábrica na qual pode haver incêndios e o vértice v_4 representa uma usina nuclear, na qual podem ocorrer vazamentos de radioatividade.

Cenário 1: Patrulha Cega

Nas abordagens de patrulhamento existentes, agentes se deslocam pela região tendo como objetivo identificar eventos. No entanto, devido à falta de conhecimento sobre a natureza dos eventos e do comportamento do ambiente, seu objetivo consiste basicamente em reduzir o intervalo de tempo entre duas visitas consecutivas a um mesmo local e assim aumentar suas chances de sucesso. Conforme vimos anteriormente, a métrica que descreve esse comportamento é denominada ociosidade¹ $I(v_i)$. Assumindo que $I(v_2) = 2$, $I(v_3) = 1$ e $I(v_4) = 0$, o agente seguirá o trajeto formado pelos vértices $\{v_1, v_2, v_3, v_4\}$, pois $I(v_2) > I(v_3) > I(v_4)$.

Dessa forma, observamos que o agente verifica inicialmente se há alguém se afogando, o que em caso de confirmação o levaria a salvar uma vida. Em seguida ele visita a fábrica a fim de verificar a ocorrência de incêndio, cujas consequências seriam desde a destruição completa das instalações até a morte de várias pessoas. Por último ele verifica se há um acidente nuclear, o qual, dependendo de sua intensidade e da densidade populacional da região, uma grande quantidade de pessoas pode ser atingida, sendo acometidas com doenças incuráveis e mutações genéticas, as quais poderão prejudicar gerações futuras.

Apesar de que, do ponto de vista da métrica escolhida, podemos afirmar que o agente obteve sucesso em sua patrulha, esse tipo de estratégia não reflete o comportamento esperado de uma patrulha no mundo real. Essa discrepância fica ainda mais clara quando consideramos a probabilidade de ocorrência de cada evento. Por exemplo, se a patrulha está sendo realizada no período noturno ou a estação é inverno, em termos gerais a probabilidade de ocorrer afogamento seria baixa, o que deveria fazer com que os agentes visitassem essa parte da cidade com menos frequência. Ainda

¹ Estamos usando aqui $I(v)$ para definir a ociosidade devido ao termo original em inglês (idleness) conforme amplamente usado na literatura de .

assim, devido à ociosidade do vértice ser mais alta, o agente irá procurar por eventos naquele vértice com prioridade máxima naquele momento.

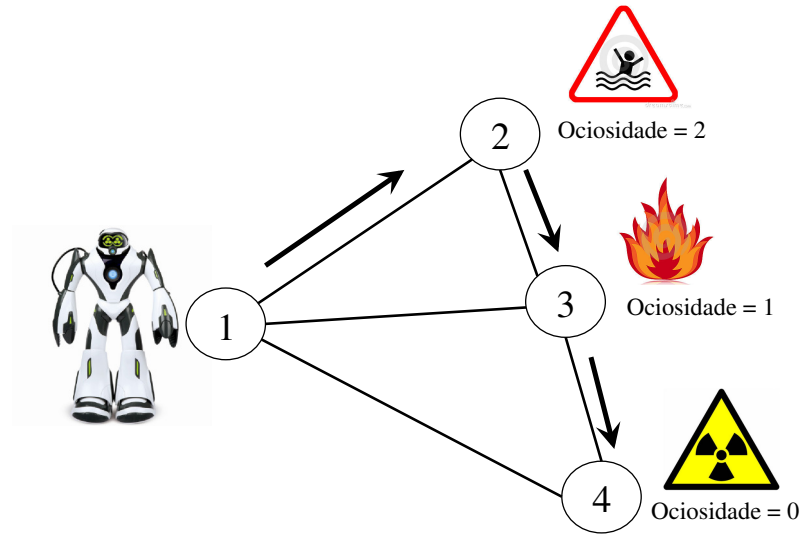


Figura 5.1: Estratégia de visita na patrulha cega.

Cenário 2: Patrulha com Conhecimento

Um sistema de patrulhamento no mundo real deve se comportar de forma diferente à apresentada no exemplo anterior a fim de ser mais eficiente e evitar ou mitigar prejuízos mais eficientemente. Para isso o agente deve decidir quais vértices serão visitados utilizando outros critérios, tais como, por exemplo, o *fator de impacto* do evento, *probabilidade de ocorrência do evento*, *distância* etc.

Nesse caso a estratégia do agente é baseada no conhecimento que ele tem à respeito da natureza dos eventos e do estado corrente do mundo. Ele deve, portanto, analisar qual tipo de evento pode ocorrer em cada vértice no instante em que for deliberar. Assumindo que ele utilizará como parâmetro o *fator de impacto*² (\mathcal{F}), que descreve a capacidade do evento em causar danos (materiais ou imateriais), primeiro o agente verifica a ocorrência de vazamentos na usina nuclear, depois a de incêndios e, por último, a ocorrência de afogamento. Conforme a figura 5.2, teremos a seguinte sequência de visitas: $\{v_1, v_4, v_3, v_2\}$.

² Esse conceito será melhor explicado no próximo capítulo.

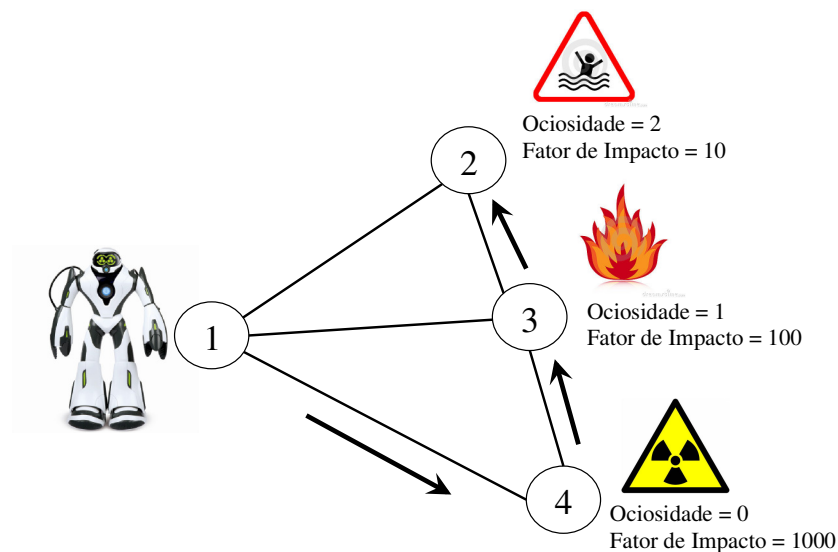


Figura 5.2: Estratégia de visita na patrulha com conhecimento.

Vamos agora assumir que cada evento tem uma probabilidade de ocorrência $p(E_i)$ de tal forma que o agente tome suas decisões sem saber a priori se o evento vai ocorrer. A decisão do agente será baseada no produto de $\mathcal{F}(E_i)$ por $p(E_i)$. Nesse caso, assumindo que afogamento, incêndio e acidente nuclear tem probabilidades $p(E_1) = 0.9$, $p(E_2) = 0.2$ e $p(E_3) = 0.01$, respectivamente, e em seguida fazendo $p(E_i) \times \mathcal{F}(E_i)$, obtemos o conjunto de resultados $\{9,0; 20,0; 10,0\}$. Dessa forma, o agente irá procurar primeiro por afogamentos, em seguida por incêndios, e, por último, por acidente nuclear, conforme ilustrado na figura 5.3.

Podemos perceber, portanto, que essa forma de enxergar o problema da patrulha tem uma grande influência sobre aplicações de monitoramento e vigilância. Uma vez que os agentes visitam os vértices de acordo com o conhecimento da natureza dos eventos, as chances de evitá-los ou minimizar as consequências serão maiores.

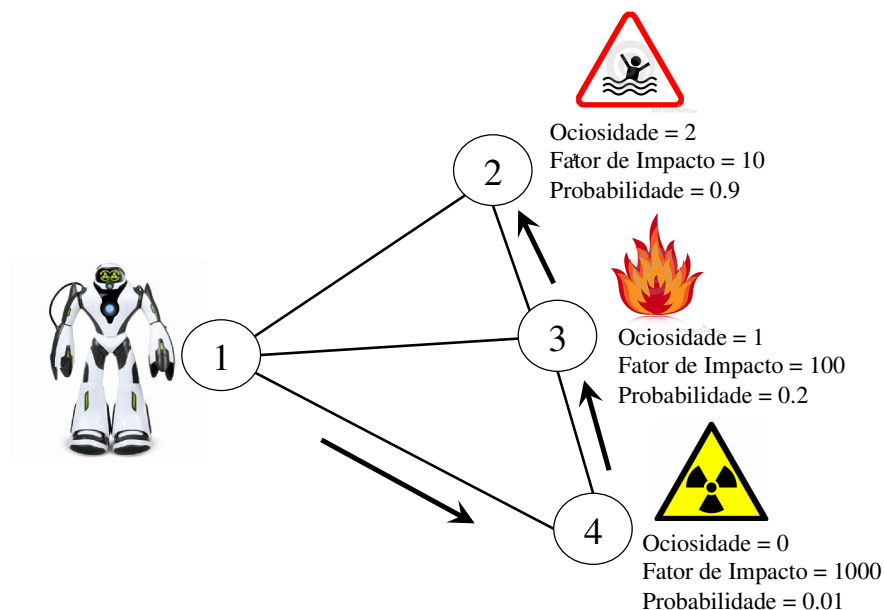


Figura 5.3: estratégia de visita baseado em fator de impacto e probabilidade de ocorrência.

5.2.5 Discussão

No presente trabalho, o objetivo dos agentes patrulhadores é identificar se há um evento ocorrendo em cada vértice visitado. Para isso ele deve utilizar seu conhecimento sobre os eventos e sobre os locais que serão visitados. Toda vez que ele conseguir identificar um evento - ou seja, tomar a decisão correta, ele será recompensado. Isso fará com que o agente busque sempre aqueles locais onde há maiores chances de recompensa.

Além disso, além de identificar em qual vértice está ocorrendo (ou vai ocorrer) determinado evento, é importante que os agentes cheguem o mais cedo possível naquele vértice. Por exemplo, no caso de “afogamento”, não é útil o agente chegar muito antes, pois não conseguirá evitar o evento, e nem após seu término, pois a pessoa já estará morta. O ideal é que ele chegue durante a ocorrência do evento. Em outros casos como “acidente nuclear” ou “explosão de caldeira” é possível monitorar os equipamentos e identificar a iminência do evento através da análise da temperatura do equipamento e conseguir evitá-lo. Por outro lado, no caso de uma colisão de automóveis é praticamente impossível evitá-lo ou identificá-lo no exato instante de ocorrência em função de sua duração instantânea. Nesse caso, o agente deve chegar o mais cedo possível após a ocorrência e manipular as consequências, acionando os serviços de urgência.

Em nosso trabalho, eventos representam incidentes, acidentes ou desastres que podem acontecer em uma determinada região, seja por meio de fenômenos naturais, de

atitudes intencionais ou acidentais, ou do mau funcionamento de equipamentos. Cada evento ocorre de acordo com o tipo e o estado corrente do vértice. Em decorrência desse fato, a prioridade dos vértices pode variar ao longo do tempo. Podemos afirmar, portanto, que nosso trabalho tem um enfoque em prover meios de estudar, desenvolver e testar agentes voltados à aplicações de segurança dentro de uma perspectiva realista.

Adicionalmente, podemos observar que esse tipo de patrulha pode ser estendida a outros tipos de aplicações onde os agentes buscam por melhores ofertas de preço e esses preços variam de acordo com eventos, tais como queda na bolsa de um país, interferências naturais, alterações na política econômica de um país etc.

Esse novo tipo de problema permitirá que outros questionamentos sejam feitos a fim de entender melhor o comportamento de agentes patrulhadores. Experimentos poderão permitir responder questões tais como as seguintes:

- Quantos agentes são necessários para patrulhar uma região com uma certa quantidade de vértices e manter um nível satisfatório de detecção de eventos?
- Os agentes estão detectando mais eventos de maior fator de impacto ou de menor?
- Qual o índice de acerto dos agentes em detectar determinados tipos de eventos?
- É mais eficiente buscar os eventos mais próximos mas com menor probabilidade ou os mais distantes com maior probabilidade?
- Qual o efeito da arquitetura na identificação dos eventos?

Cada pergunta poderá ser explorada mediante alterações nos atributos do agente e/ou do cenário (por exemplo, o tamanho da sociedade). Para isso o problema deve ser descrito por meio de atributos que possam ser modificados.

Um aspecto importante em relação à adoção do conceito de evento no problema da patrulha é que, além de possibilitar a realização de pesquisas que não podem ser feitas utilizando os tipos de patrulhas existentes (MATP e AMAP), PMAOE permite a unificação das abordagens, simplificando sua compreensão e aumentando o seu poder explanatório.

Na próxima seção iremos apresentar a definição de evento de acordo com o problema da patrulha.

5.2.6 Definição de Evento no Contexto da Patrulha

Essa definição visa preencher uma lacuna existente nos trabalhos que tratam sobre a patrulha, que é a ausência de um entendimento claro, abrangente e coerente sobre o que

é evento. Apesar de que este termo é utilizado informalmente na literatura (Machado et al, 2002; Almeida et al, 2003; Almeida et al, 2004; Santana et al; 2004; Elmaliach et al., 2007), faz-se necessário desenvolver uma específica para o contexto deste problema a fim de sermos mais precisos na proposição de nosso benchmark. Para tanto, elaboramos a seguinte definição:

Definição 2 (Evento): *É um acontecimento passível de identificação por um agente que ocorre em um determinado lugar e momento, sob certas circunstâncias de causalidade e a partir do qual haverá uma ou mais consequências.*

Essa definição parte da premissa que *evento é um acontecimento* (Bennet, 1988). Uma vez que, do ponto de vista da metafísica, evento *é algo que ocorre*, diferente de outras categorias, tais como objetos, que existem ou tem duração, podemos afirmar que ele tem uma localização no *espaço* e um *instante no tempo* (Kim, 1993). Um evento é definido como sendo uma classe ou categoria através da qual, particulares ou instâncias são criadas. Um evento do tipo “acidente nuclear” pode ter como instâncias o vazamento na Usina de *Three Miles Island* (EUA), *Chernobyl* (Rússia) e *Fukushima* (Japão).

A fim de esclarecer quais tipos específicos de eventos serão tratados no problema da patrulha em PMAOE, é preciso dizer que os eventos que estamos interessados ocorrem no mundo exterior à nossa mente, retirando de nosso modelo qualquer tipo de *evento mental*, tais como aqueles presentes na Filosofia da Mente, Psicologia ou Psiquiatria: reconhecimento, satisfação, lembrança, negação etc. (Davidson, 1987). Portanto, estamos preocupados apenas com eventos que podem ser percebidos por um agente a partir de uma percepção do estado do mundo em que ele está inserido para que ele possa tomar alguma decisão e realizar alguma interferência a fim de manter sua segurança, seja evitando ou minimizando os prejuízos decorrentes.

As noções de *espaço* e *tempo* desempenham um papel fundamental tanto na habilidade do agente em identificar eventos como de caracterização de cada uma das instâncias dos eventos. Quando dizemos que dois eventos *ocorrem* no mesmo espaço e tempo, significa dizer duas coisas:

- **Quanto ao espaço:** o lugar onde os eventos ocorrem se refere a um mesmo vértice dentro do grafo sendo patrulhado. Se um vértice representa um prédio (edifício ou residência), então podem ocorrer no mesmo instante dois eventos não relacionados entre si, tais como incêndio e explosão;
- **Quanto ao tempo:** a ocorrência de dois eventos pode ser considerada totalmente concorrente ou parcialmente concorrente. Considerando que o

tempo seja organizado de forma discreta, dois eventos e_i e e_j quaisquer, podem ocorrer de três formas: dois eventos começam e terminam no mesmo instante (e_1 e e_2), dois eventos começam no mesmo instante, mas um deles termina antes (e_3 e e_4), e um evento inicia e após algum tempo o outro inicia e terminarem no mesmo instante (e_5 e e_6). Essas situações são ilustradas na figura 5.4.

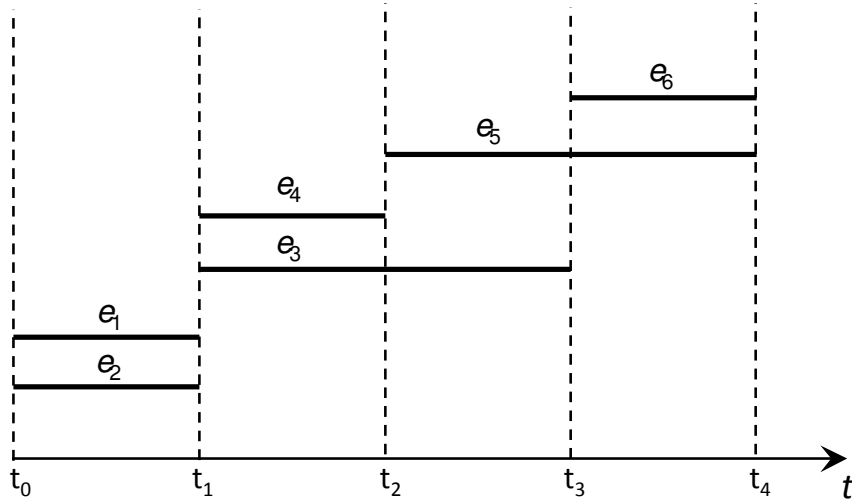


Figura 5.4: Relação temporal entre eventos.

A fim de tornar a definição de evento no contexto da patrulha mais clara e completa, desenvolvemos as seguintes propriedades:

Propriedade 1: *Um evento tem uma duração e pode se propagar no espaço.*

Todo evento ocorre em um determinado instante de tempo e pode ter ou não uma duração significativa. A duração do evento pode ser tão curta que o agente não tem capacidade de percebê-lo ou realizar qualquer ação válida durante sua ocorrência, mas somente depois, no sentido de mitigar as consequências. Por exemplo, a duração de uma colisão entre dois automóveis para um agente patrulhador é totalmente desprezível. Por outro lado, um evento pode ter uma longa duração, como por exemplo o vazamento de radiação em uma usina.

Um evento pode se expandir geograficamente em relação à sua origem. Isso significa que o evento pode abranger desde um pequeno espaço, como por exemplo uma colisão, até grandes regiões tais como continentes inteiros, como foi o caso do acidente na Usina de Chernobyl (INSAG, 1992).

Propriedade 2: *Eventos ocorrem independente da participação ou testemunho de agentes.*

Os tipos de eventos de nosso interesse não dependem exclusivamente de um agente humano para existir, ou seja, tê-lo como causador, testemunha ou vítima. Em outras palavras, o evento ocorre no mundo exterior e não representa um fenômeno mental (Davidson, 1987). Por exemplo, um incêndio florestal pode ocorrer devido à combustão em decorrência da vegetação estar bastante seca. No entanto, esse evento deve ser passível de identificação por parte dos agentes.

Propriedade 3: *Instâncias de eventos diferem entre si por alguma propriedade.*

De acordo com Kim (1993), instâncias de eventos são particulares irrepetíveis. Davidson (1980) cita a temporalidade como requisito para a individuação. No entanto, essas afirmações consideram apenas a temporalidade como elemento de individuação somente quando eventos ocorrem em instantes diferentes. Por exemplo, em uma cidade podem ocorrer ao mesmo tempo dois atentados diferentes, no mesmo instante mas em lugares diferentes. Sendo assim, além do tempo, o espaço também é condição necessária para diferenciar um evento de outro.

Instante e local de ocorrência são elementos diferenciadores entre dois eventos em nossa abordagem conceitual, mas não suficientes. Assim, dois eventos podem ser idênticos, mas não iguais em termos de sua identidade. Cada tipo de evento E_i é definido por meio de um conjunto de propriedades p e define uma classe de ocorrências de eventos $\{e_1, e_2, \dots, e_n\}$. Cada instância de evento é individual no sentido de que mesmo que eles ocorram no mesmo instante e local, existe a possibilidade de identificá-los isoladamente. Mesmo que $p(e_1) = p(e_2)$, as duas instâncias serão diferentes. Portanto, em nossa abordagem uma instância de um evento é individual e irrepetível.

Propriedade 4: *Todo evento tem uma ou mais causas e uma ou mais consequências e depende do estado atual do local onde ocorre.*

A ocorrência de um evento demanda que eventos causais ocorram anteriormente dependendo de certas condições antecedentes, de acordo com as leis da natureza (Hofer, 2010). Por exemplo, o evento “incêndio” sucede a ocorrência do evento “acender fósforo” (ou “produzir fagulha”) e depende das condições “*existe material inflamável no local*” e “*existe oxigênio no local*” serem verdadeiras. Dada a ocorrência desse evento e a existência dessas condições, teremos o evento “incêndio”, o qual será sucedido pelos eventos “aumento da temperatura ambiente”, “formação de gases”, “destruição de matéria” etc.

Conforme pode ser visto na figura 4.4, eventos podem ocorrer devido a um ou mais eventos causais simultâneos. Por exemplo, os eventos “*carro passa à frente*” e “*freio não funciona*” antecedem o evento “*colisão de automóveis*”. Similarmente, eventos podem levar a um ou mais eventos consequentes simultâneos. Nesse caso, uma colisão pode causar os eventos “*deformação na estrutura dos automóveis*”, “*ferimento nos passageiros*”, “*destruição de postes de iluminação*” etc.

No esquema conceitual que apresentamos nesse trabalho para lidar com eventos, faz-se necessário fazermos uma importante simplificação no que diz respeito à estrutura de um evento. Para isso a limitamos como sendo constituída às três partes anteriormente discutidas. No entanto, um evento causal, por definição, é também um evento, o que, do ponto da metafísica, impõe a necessidade de termos um evento causal para ele e assim sucessivamente. A fim de concentrarmos nossa atenção sobre um intervalo de tempo razoável em torno do evento principal, restringiremos a percepção do agente e a influência do evento principal em termos de um ou mais eventos causais (desde que simultâneos) e ou mais eventos consequência (desde que simultâneos).

Além dessas propriedades, eventos são estruturados de acordo com um conjunto de atributos conforme será apresentado na próxima seção. Para isso nós desenvolvemos uma taxonomia, elencando suas principais características, a qual permite caracterizar de forma clara, consistente e abrangente cada tipo de evento e assim definir os cenários.

5.3 Modelagem da Classe de Problema

5.3.1 Justificativa

Conforme visto no capítulo 3, um entendimento detalhado do problema da patrulha deve ser fornecido a fim de torná-lo um benchmark capaz de atender às necessidades de pesquisa em SMA. Apesar de que existe um entendimento informal do que consiste a patrulha, uma grande quantidade de detalhes precisa ser esclarecida e melhor definida. Para isso devemos estender a compreensão do problema além da definição informal largamente utilizada na literatura. Nesse caso é importante dispormos de um modelo que descreva quais são seus elementos básicos. Dessa forma desenvolvemos uma taxonomia com o objetivo de reunir todas as patrulhas sob uma mesma categoria ontológica. A taxonomia nos fornece vantagens tais como:

- **Padronização do conhecimento:** conforme vimos no capítulo anterior, um dos problemas em é a falta de uma definição comum e rígida. Uma taxonomia fornece uma estrutura conceitual comum que pode servir de

referência para o aprendizado, desenvolvimento e comparação de trabalhos realizados por diferentes grupos de pesquisa. Sendo assim, a taxonomia deve ser desenvolvida de tal forma que represente o conhecimento necessário para descrever qualquer tipo de patrulha;

- **Simplificar o entendimento de** : apesar da aparente simplicidade do problema, a adoção de diferentes caracterizações do problema em alguns trabalhos, bem como a ausência de informações em outros, prejudica seu entendimento. Portanto, é essencial reunir e apresentar de forma estruturada quais são os elementos fundamentais de e como eles se relacionam;
- **Aumentar a produtividade:** a desorganização conceitual de faz com que a produtividade dos pesquisadores seja prejudicada, dificultando sua adoção com benchmark em SMA. Dentre alguns problemas podemos citar a dificuldade de comunicação, conhecimento desestruturado e falta de reuso de experiência;
- **Facilitar a divulgação de resultados:** a existência de um vocabulário compartilhado facilita a compreensão dos trabalhos científicos bem como qualquer outra forma de comunicação no sentido de divulgar a pesquisa sendo desenvolvida no contexto de . Sabendo quais elementos são fundamentais em uma instância de um problema, pesquisadores poderão abordá-los de forma padronizada;
- **Possibilitar a comparabilidade de resultados:** para que duas ou mais soluções possam ser comparadas, é preciso que ambas adotem a mesma definição de problema ou que estas sejam compatíveis em torno das variáveis críticas³. Isso é fundamental na compreensão do quanto se avançou em determinada questão. Por exemplo, as conclusões obtidas a partir da comparação entre um trabalho abordando grafos circulares com agentes dispostos uniformemente (Agmon et al., 2009) e outro abordando grafos em ilhas com agentes dispostos não uniformemente (Almeida et al., 2002) são limitadas devido à diferença de parâmetros que afetam a compreensão dos resultados;
- **Simplificar a especificação de instâncias de** : a taxonomia representa variações de . Dessa forma será possível criar instâncias através da atribuição de valores a cada atributo da taxonomia. Conforme veremos na próxima seção, essa abordagem permite que uma quantidade vasta de combinações seja obtida, muitas delas sem aplicação na prática. Adicionalmente, a falta de um conjunto comum de parâmetros torna

³ Por variável crítica queremos dizer aquelas que interferem diretamente no resultado da solução.

complexa a tarefa de identificar instâncias de problemas relevantes à realização de pesquisas;

- **Classificar os trabalhos existentes:** a taxonomia deve permitir a classificação dos trabalhos existentes de acordo com suas características básicas. Por exemplos, no que diz respeito à região patrulhada deve ser interessante identificar quais trabalhos lidam com determinado tipo de grafo;
- **Identificar oportunidades de pesquisa:** uma vez que a taxonomia permitirá a classificação dos trabalhos desenvolvidos, consideramos que através dela será possível também descobrir se alguma característica em particular da patrulha não foi investigada. Dessa forma, pesquisadores poderão concentrar seus esforços de pesquisa a partir de uma análise sistemática da literatura existente tendo como base a taxonomia;

5.3.2 Classificação das Patrulhas Existentes

Antes de definir a estrutura conceitual e terminológica de e dos eventos, é importante entender a relação entre os tipos de patrulhas existentes. A questão é que apesar de já haver uma quantidade significativa de trabalhos envolvendo o problema da patrulha, não existe uma organização conceitual das diversas abordagens existentes. Por exemplo, não existe nenhuma clareza a respeito da forma com que cada um dos trabalhos se relaciona com os demais, quais são as características fundamentais de uma patrulha, se existem características que permitam agrupá-las sob uma mesma perspectiva, qual é o tipo de cada patrulha desenvolvida etc. Dessa maneira, propomos uma classificação dos trabalhos da patrulha de forma a facilitar tanto o entendimento como o projeto de soluções baseadas em .

Essa classificação é baseada no conceito de evento. De acordo com ela, existem três tipos de patrulhas (figura 5.1): AMAP (*Adversarial MultiAgent Patrolling*), PMAOE (*Event Oriented Multiagent Patrolling*), MATP (*MultiAgent Timed Patrolling*). Ela parte do princípio que toda patrulha tem como objetivo identificar eventos. Nos trabalhos que lidam com invasão, os agentes patrulhadores percorrem os diversos pontos da região a fim de detectar agentes invasores. Nesse caso, o evento que o agente está procurando consiste em uma invasão.

Em AMAP, existem dois tipos de agentes: *agente supervisor* e *agente invasor*. O objetivo do agente supervisor (ou patrulhador) é identificar a presença de agentes que planejam invadir o grafo ou visitar vértices que possuem algum valor de interesse ao invasor, tais como lojas, residências, agências bancárias ou servidores de rede (Paruchuri et al., 2006; Paruchuri et al; 2007a; Paruchuri et al; 2007b; Gatti, 2008; Paruchuri et al; 2008; Agmon et al., 2009; Basilico et al; 2009a; Basilico et al; 2009b;

Basilico et al; 2009c; Marino et al., 2009; Marino et al., 2010; Basilico et al., 2010;). O objetivo do agente invasor (também chamado de adversário) é identificar quais vértices apresentam menor probabilidade de ser visitado pelo agente supervisor. Para isso ele, eventualmente, pode dispor de um modelo do comportamento do agente patrulhador.

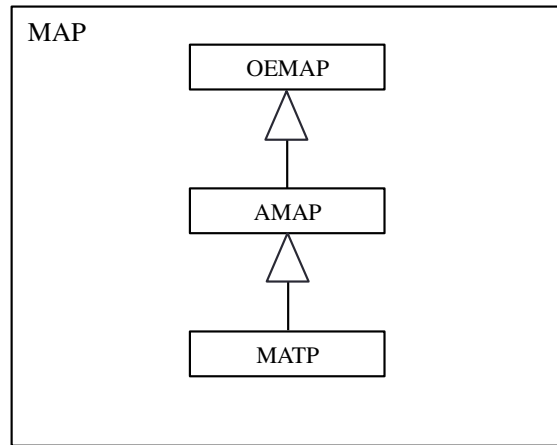


Figura 5.5: Relação entre tipos de patrulhas em .

Em MATP os agentes patrulham a região em busca de qualquer tipo de evento. No entanto, eles não tem conhecimento explícito dos eventos que ocorrem no mundo, de tal forma que a deliberação acerca de qual vértice deve ser visitado se baseia principalmente no *tempo*⁴. Dessa forma, eles procuram minimizar o intervalo de tempo entre duas visitas consecutivas a fim de coincidir com o momento de ocorrência do evento. Em outras palavras, os agentes baseiam suas decisões observando os instantes em que os vértices foram visitados pela última vez, mas sem possuir qualquer informação acerca da natureza dos eventos (Machado et al., 2002; Almeida et al., 2003; Sempé e Drogoul, 2003; Almeida et al., 2004; Chevaleyre 2004; Santana et al, 2004; Chu et al., 2006 a, Chu et al., 2006 b; Elmaliach et al; 2006; Menezes et al; 2006; Moreira, 2008; Sak et al, 2008; Portugal, 2009; Portugal, 2010; Sampaio et al., 2010). Podemos, portanto, afirmar que o agente faz uma espécie de *busca cega* a fim de localizar o ponto de interesse onde um evento poderá estar ocorrendo. É importante ressaltar que os trabalhos que utilizam essa abordagem se baseiam em técnicas de otimização como forma de reduzir ao máximo a frequência de visitas aos vértices.

Acerca da relação entre essas classes de patrulhas, duas observações devem ser feitas: a primeira é que o que muda entre AMAP e MATP é o fato de que, no primeiro

⁴ Em alguns trabalhos como o de Almeida e colegas (2003) são usados outros critérios tais como a distância entre o vértice de origem e o de destino na escolha dos vértices a serem visitados.

caso, existe um agente invasor que se move em direção a pontos específicos da região. No entanto, como o agente patrulhador desconhece sua existência, ele adota a mesma estratégia de minimizar a ociosidade dos vértices de MATP. Podemos afirmar, portanto, que AMAP é uma subclasse da MATP. A segunda observação é que, como em MATP não é explícito o que se deseja identificar, é assumido que se trata de um evento qualquer (uma invasão, vazamento de gás, afogamento etc.) que deve ser identificado, sem se levar em consideração qualquer característica sobre ele. Portanto, MATP é, na verdade, uma estratégia adotada devido à falta de informações sobre o ambiente, não correspondendo de fato a um problema do mundo real.

Por fim, temos PMAOE (Agmon, 2010). Nesse tipo de patrulha os agentes dispõem de conhecimento explícito acerca dos eventos (Mazda e Stone, 2006; Agmon et al., 2009; Agmon, 2010). Apesar de ser um cenário promissor para , a forma como o conceito de evento é utilizado é bastante limitada: além de ser intuitivo (“algo que acontece”), não leva em consideração as suas diversas particularidades. Nessa tese propomos que o agente disponha de conhecimento detalhado acerca dos eventos tais como causa, consequência, duração etc. Portanto, PMAOE representa a idéia de patrulhamento de forma ampla: o agente sabe quais são os locais da região onde e quando os eventos ocorrem com maior probabilidade. Uma vez que, conforme discutido acima, uma invasão é um tipo de evento, podemos afirmar que AMAP é uma subclasse de PMAOE.

5.3.3 Taxonomia da Classe de Problema

A taxonomia agrupa o conhecimento de em cinco dimensões principais, conforme pode ser visto na figura 5.6: *grafo*, *sociedade*, *agente*, *tempo de simulação* e *métricas*. O desenvolvimento dessa taxonomia foi realizado por meio de uma análise da literatura especializada e da experiência do grupo de pesquisas em do CIn-UFPE. Nas subseções seguintes iremos detalhar cada uma dessas delas.

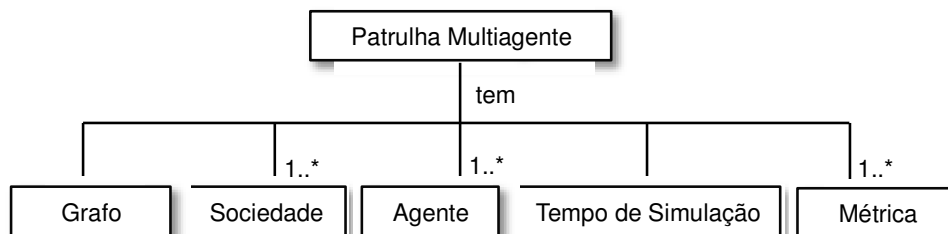


Figura 5.6: Visão de alto nível da Patrulha Multiagente.

Grafo

Grafo é uma abstração do terreno onde o agente está patrulhando, conforme apresentado no capítulo anterior. O terreno é geralmente representado como um grafo onde os vértices correspondem aos lugares a serem visitados e os arcos aos caminhos existentes entre eles. A fim de caracterizar um problema, um grafo deve descrever os seguintes elementos básicos: *tipo*, *vértice*, *arco* e *tamanho* (figura 5.7).

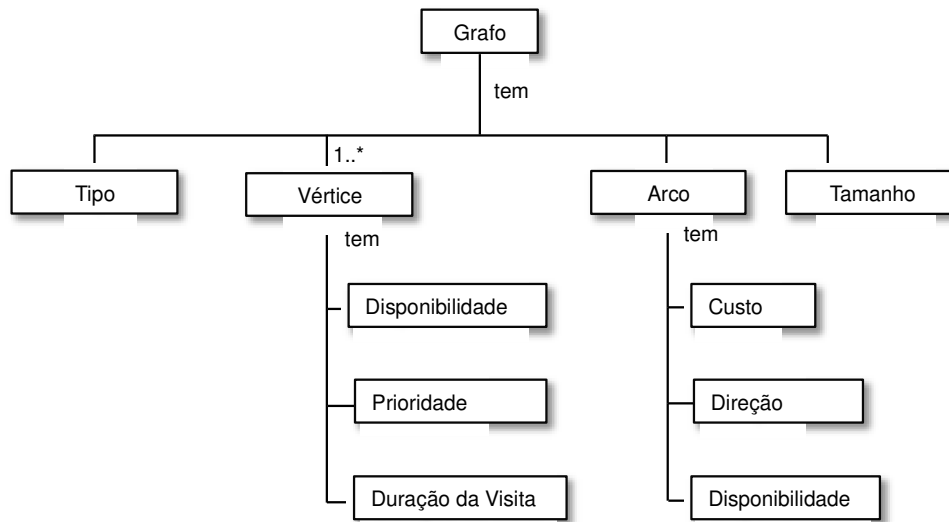


Figura 5.7: Caracterização de um grafo no problema .

Tipo se refere à estrutura espacial ou topologia do grafo. É importante ressaltar que até então não existe nenhum entendimento claro ou profundo das características dos elementos que melhor descrevem a topologia de um grafo em termos de seu impacto para o problema, mas experimentalmente sabemos que algumas características que este impacto é relevante para o problema (Almeida et al., 2004). Informalmente temos observado que algumas características são importantes. Por exemplo, é mais fácil patrulhar em grafos com uma maior densidade de arcos (grafos altamente conectados) e uma distribuição espacial uniforme dos vértices assim como grafos contendo ciclos. Apesar da falta de formalização, o uso de diferentes topologias, tais como grades, ilhas, corredores e círculo, ajudam os pesquisadores a melhor avaliar a generalidade de suas abordagens.

Vértice representa os locais visitados para fins de patrulhamento em uma determinada região, tais como residências, bancos, máquinas, etc. Possui as seguintes propriedades: *dinamicidade*, *prioridade* e *duração da visita*.

- **Dinamicidade** refere-se à possibilidade de um vértice já existente ser removido do grafo ou de um novo ser adicionado. Isto corresponde a situações do mundo real onde os locais a serem visitados não estão mais acessíveis ou simplesmente não existem mais, ou novos locais são criados dentro da região a ser patrulhada. Por exemplo, uma determinada agência bancária encerra suas atividades definitivamente, não sendo mais necessário monitorá-la.
- **Prioridade** refere-se à importância de um determinado vértice em relação aos demais, o que fará com que o agente decida por visitá-lo em detrimento dos demais que possuem prioridade menor. Isso é importante para descrever situações em que podem ocorrer eventos de diferentes tipos nos vértices, cada um com consequências variadas. Nesse caso, é fundamental que o agente possa decidir visitar aquele vértice em que existe a possibilidade de haver riscos às pessoas lá localizadas do que visitar aqueles em que existem apenas bens materiais.
- **Duração** da visita define a quantidade de tempo em que os agentes permanecem em um determinado vértice.

Arco representa o caminho que liga um vértice a outro e é caracterizado pelos seguintes atributos: *custo*, *direção* e *dinamicidade*.

- **Custo** representa o esforço necessário para o agente ir de um vértice a outro por meio de um determinado arco. Esse custo pode ser um valor financeiro a ser pago, o tempo necessário para atravessá-lo, a quantidade de energia a ser consumida, etc.
- **Direção** define a forma como os agentes podem atravessar os arcos de um grafo. Isto é importante não apenas porque ele tem uma associação direta com aplicações do mundo real (por exemplo, automóveis em uma cidade), mas porque tem um impacto na conectividade do grafo e, conseqüentemente, no desempenho da patrulha.
- **Disponibilidade** do arco é semelhante à do vértice, com basicamente o mesmo impacto. Entretanto, se um arco for removido, o agente poderá alcançar o destino utilizando um outro caminho alternativo. A consequência de se remover um vértice pode ser maior que no caso de um arco.

Tamanho de um grafo descreve quantos vértices o grafo possui. Ele permite saber, por exemplo, qual a distribuição dos agentes no grafo, ou seja quantos vértices podem ser alocados para cada agente.

Sociedade

Esta dimensão descreve as características da sociedade de agentes no que diz respeito ao problema da . Ela tem quatro propriedades, conforme pode ser visto na figura 5.8: *organização, dinamicidade, comunicação e tamanho*.

Organização diz respeito ao fato de que uma sociedade é composta por agentes com as mesmas habilidades, obrigações e papéis. No caso afirmativo, dizemos que a sociedade é *homogênea* e do contrário, que é *heterogênea* (Russel e Norvig, 2004; Wooldridge, 2010).

Dinamicidade está relacionada com a capacidade do tamanho da sociedade aumentar ou diminuir durante a patrulha. Em outras palavras, o agente pode entrar ou deixar a sociedade. É importante ressaltar que a sociedade ser aberta não é uma decisão de projeto, mas uma imposição do problema. Por exemplo, patrulhamento militar e/ou baseado em robôs móveis frequentemente são sociedades abertas.

Comunicação refere-se à habilidade dos agentes trocarem informações entre si. Em alguns problemas, os agentes não se comunicam, mas em outros eles podem se comunicar diretamente, por meio de mensagens, ou indiretamente, usando a técnica do quadro-negro ou de bandeiras (Almeida, 2002).

Tamanho refere-se à quantidade de agentes que uma determinada sociedade possui. Esta característica tem um importante impacto na escalabilidade da solução proposta.

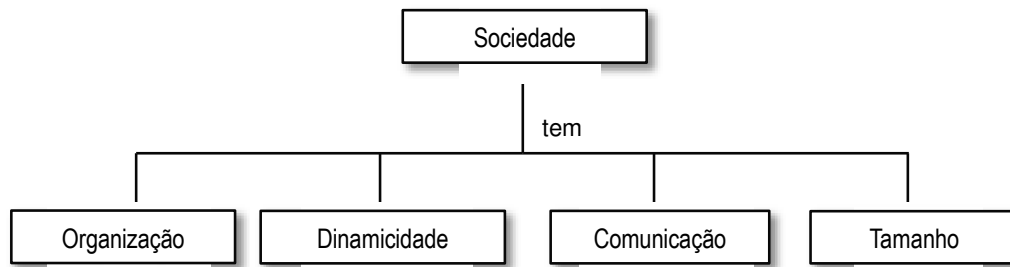


Figura 5.8: Caracterização de uma sociedade no problema da patrulha.

Agente Patrulhador

Esta dimensão descreve as principais características da arquitetura do agente em termos das restrições impostas pelo problema . Ela é composta das seguintes características, conforme ilustrado na figura 5.9: *informação a priori e percepção*.

Informação a priori se refere ao conjunto de informações que o agente dispõe sobre o ambiente antes de iniciar a patrulha. Nesse caso, o agente pode ter informações sobre a sociedade e sobre o grafo.

- **Sociedade** permite ao agente saber duas informações a respeito da sociedade da qual ele faz parte: dinamicidade, ou seja se a sociedade é aberta ou não, e o seu tamanho;
- **Grafo** indica se o agente dispõe ou não de informações sobre o grafo. Assim, ele poderá saber a priori a dinamicidade e a prioridade de cada vértice;

Percepção representa a capacidade do agente perceber o ambiente durante a patrulha, e, assim, tomar decisões a respeito de quais são os vértices mais interessantes em visitar. Essa característica descreve o *alcance de sua percepção* e o que pode ser percebido, i.e., *o objeto da percepção*.

- **Objeto** representa tudo o que pode ser percebido pelo agente em um ciclo de percepção: estado atual do grafo (quantos arcos o grafo possui naquele instante, o tipo de cada vértice vizinho), sociedade (quantos agentes existem no grafo naquele momento), tempo (instante de tempo corrente, última visita realizada à cada vértice vizinho), custo do arco (distância, quantidade de energia necessária para atravessar ou quantidade de ciclos) e evento (quais e quantos eventos são percebidos a cada percepção).
- **Alcance** representa a profundidade da percepção do agente. O agente pode enxergar apenas os vértices diretamente conectados (percepção local), todos os vértices do grafo (percepção global) ou valores intermediários.
- **Confiabilidade** representa a possibilidade de haver ou não ruídos na percepção.

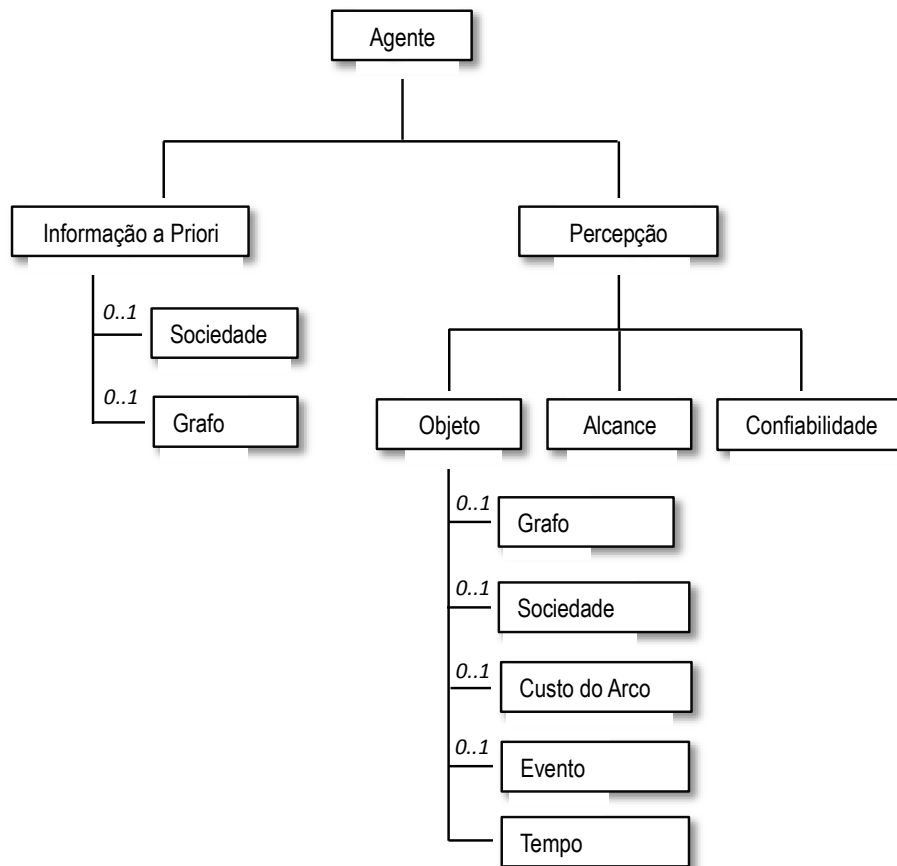


Figura 5.9: Representação de um agente patrulhador.

Métricas

Essa dimensão descreve os tipos de métricas que podem ser utilizadas na , conforme figura 5.10. Temos então os seguintes: *baseadas em tempo*, *baseadas em esforço* e *baseadas em eventos*.

- **Baseadas em Tempo:** Esse tipo de métrica visa permitir a análise do desempenho dos agentes a partir da sua capacidade de realizar a tarefa de visitar os vértices dentro do menor intervalo de tempo possível. Em outras palavras, o agente deve minimizar o intervalo de tempo entre duas visitas consecutivas em todos os vértices com o intuito de maximizar suas chances de sucesso;
- **Baseadas em Utilidade:** Nesse tipo de métrica calcula-se o quanto o agente recebeu em termos de recompensa toda vez que ele atinge um

certo objetivo (p. ex. identificar um invasor) ou o quanto ele gasta em termos de, por exemplo, energia durante toda a simulação;

- **Baseadas em Eventos:** Esse tipo de métrica visa permitir a análise do desempenho dos agentes a partir da sua capacidade de identificar eventos.

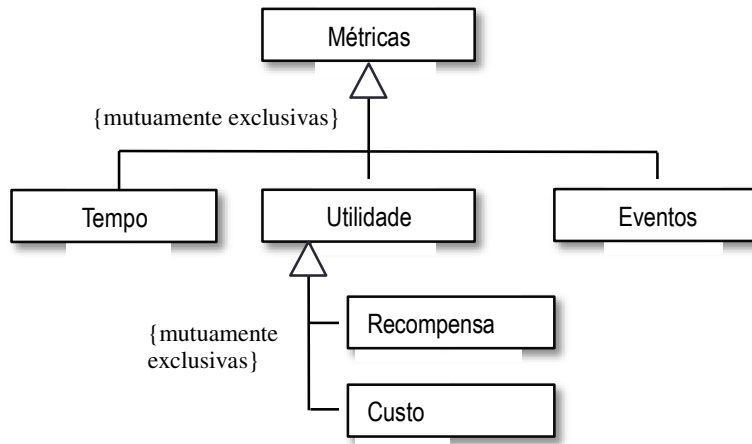


Figura 5.10: Caracterização das métricas em .

5.3.4 Exemplos de Eventos

Os eventos analisados a fim de criar a taxonomia foram os seguintes: *afogamento*, *atentado terrorista*, *incêndio*, *assalto*, *vandalismo*, *colisão de automóveis*, *pane em equipamento* e *acidente nuclear*. Eles foram escolhidos levando em consideração o fato de serem eventos tipicamente danosos e que são de grande interesse em aplicações de vigilância e monitoração, tais como aquelas estudadas na gestão de desastres ou segurança pessoal e patrimonial. Nos próximos parágrafos iremos detalhá-los individualmente.

Antes de apresentá-la é importante ressaltar o seguinte: O principal critério para a escolha dos eventos no presente estudo é a possibilidade dos mesmos virem a comprometer a segurança tanto das pessoas como dos demais recursos localizados em uma determinada região. Considerando que o ato de patrulhamento tem como objetivo a vigilância e monitoramento, consideramos que esse critério é fundamental na definição dos atributos que caracterizam os eventos para a PMAOE;

Afogamento: é o processo decorrente da diminuição da respiração a partir da imersão ou submersão em líquido (van Beek et al., 2005). Segundo Idris e colegas (2003), havendo resgate (interrupção do afogamento), a vítima pode se recuperar com ou sem terapia para eliminar a hipoxia, hipercarbina e acidose, e o funcionamento dos órgãos ser restaurado. No entanto, se a vítima não começar a respirar (por meio de

auxílio ou espontaneamente), pode haver disfunção de múltiplos órgãos (como por exemplo, a morte de células nervosas) ou até mesmo a morte da pessoa. Portanto, um afogamento pode incorrer em três situações que dependem do intervalo de tempo em que a vítima é atendida: salvamento sem danos, salvamento com danos, e morte.

Atentado terrorista: para o Departamento de Defesa dos EUA, “o termo terrorismo significa violência premeditada, motivada politicamente e perpetrada contra alvos não-combatentes por grupos nacionais ou agentes clandestinos, geralmente com a intenção de influenciar uma audiência” (Military Periscope, 2004). As informações de que os responsáveis pelos ataques terroristas agem de forma planejada, clandestina (escondida) e possuem alvos preferenciais (aqueles que possibilitem maior repercussão), são de grande importância para a compreensão de como se dá seu comportamento e para a constatação de que, em muitos casos, são de difícil detecção. Do ponto de vista da patrulha esse tipo de evento é difícil de evitar ou lidar no exato momento em que ocorre, restando aos agentes a alternativa de identificá-lo o mais rápido possível após sua ocorrência.

Incêndio é um evento causado pela propagação ou desenvolvimento com grande intensidade de fogo, causando prejuízos materiais, danos à saúde ou perda de vidas. Sua origem pode ser intencional ou acidental. Ele pode ocorrer de forma deliberada ou por meio de eventos naturais, tais como relâmpagos, incidência de raios solares sobre vegetação seca, problemas em instalações elétricas, super-aquecimento de equipamentos em fábricas etc. Um aspecto importante desse tipo de evento é que sua duração pode ser indefinida: uma vez iniciado, se ele não for interrompido, enquanto houver meios para se realizar a combustão (oxigênio e material inflamável), o incêndio tende a se alastrar e ampliar seu alcance geográfico. É importante que esse seja evitado logo que seja percebida sua iminência ou detectado o mais rápido possível após o início. Em nosso trabalho, iremos lidar apenas com incêndios acidentais, e os intencionais serão tratados como sendo vandalismo ou terrorismo. Esse tipo de evento é evitável, desde que os agentes sejam capazes de identificar os eventos causadores. Ele também é manipulável uma vez que os agentes podem identificá-lo durante sua ocorrência. Caso os agentes não consigam evitá-lo, o ideal é identificá-lo o mais cedo possível após o início de sua ocorrência.

Assalto pode ser definido como sendo o ato doloso de retirar dinheiro, bens pessoais, ou qualquer outro artigo de valor na posse da vítima, contra sua vontade e realizado por meio da força ou medo (Craighead, 2004). Nesse caso há uma abordagem direta por parte do assaltante à vítima. Isso é diferente, por exemplo, da situação em que uma pessoa invade uma residência para roubar qualquer bem material. Não será considerado para fins desse estudo outros eventos relacionados com o assalto, como por exemplo o latrocínio ou o sequestro. Nestas condições torna-se inviável para a patrulha

de agentes evitá-lo. Sendo assim, resta ao agente identificá-lo no momento em que estiver ocorrendo.

Vandalismo é uma ação dolosa com o objetivo de danificar ou destruir propriedades públicas ou particulares (Craighead, 2004). Esse tipo de ação é cada vez mais comum nos centros urbanos, principalmente nas de médio e grande porte, causando grandes prejuízos para as autoridades públicas. O comportamento do vandalismo do ponto de vista do patrulhamento é igual ao do assalto.

Colisão de automóveis é um evento que consiste em uma colisão ocasionada por automóvel, podendo envolver outros veículo, um poste de iluminação, construções, árvores etc. Uma colisão pode ocorrer por diversas razões, tais como falta de atenção do motorista, problemas no automóvel, falta de sinalização, problemas na pista etc. A colisão não pode ser evitada ou manipulada, devendo o agente identificá-la o mais rápido possível após sua ocorrência.

Acidente nuclear: esse tipo de evento pode ocorrer por diversas razões (INSAG, 1992). No entanto, por questões de simplicidade, iremos assumir uma específica: não está havendo resfriamento das turbinas do reator. Nesse caso, o agente pode monitorar a temperatura e de acordo com os seu valor, identificar ou não a iminência do evento e assim evitá-lo.

A partir da análise desses tipos de eventos e do conhecimento da Teoria dos Eventos desenvolvemos a taxonomia apresentada na figura 5.6. Nas próximas subseções iremos descrever cada um dos elementos que a compõem.

5.3.5 Taxonomia de Eventos

Evento é um conceito essencialmente abstrato e utilizado em diversas áreas e contextos. A fim de viabilizar sua utilização no problema da patrulha, é fundamental fornecer uma descrição estruturada, por meio de seus elementos básicos e de seu comportamento. Essa descrição deve ser coerente com as características da patrulha e capturar os elementos necessários tanto para a simulação de cenários como do comportamento de agentes que não podem ser estudados com os recursos atuais.

O objetivo desta seção é descrever a taxonomia de eventos conforme apresentada na figura 5.11. O seu desenvolvimento foi realizado a partir de pesquisa na literatura de teoria dos eventos, conforme visto na seção 4.6 (Davidson, 1980; Kim, 1993; Brand, 1997; Garret, 2006; Carati e Varzi, 2010; Scheneider, 2010) e pelo estudo da literatura e análise de diversos tipos de eventos. Não pretendemos abordar todas as categorias de um mesmo tipo de evento. Por exemplo, existem diversas formas de se realizar um ato terrorista (Purpura, 2006), mas por questões de simplicidade trataremos este e os demais eventos de forma genérica. A importância de definir e, principalmente, descrever um conjunto representativo de eventos é essencial para a elaboração de uma

solução que atenda adequadamente aos requisitos da PMAOE. Isso se faz necessário devido ao fato de que alguns termos têm sido usados em tantos contextos diferentes que passaram a ter significados bastante diferentes.

Nos próximos parágrafos iremos descrever cada um desses elementos:

Tipo: permite descrever o tipo de evento. Exemplos de tipo de evento são os mencionados anteriormente (“afogamento”, “assalto”, “colisão de automóveis” etc).

Categoria: permite classificar o evento quanto à sua relação com outros eventos do ponto de vista da causalidade. No contexto desse trabalho, um evento pode ser uma *causa*, um evento *principal* ou uma *consequência* de acordo com a propriedade 4 na seção 5.8.

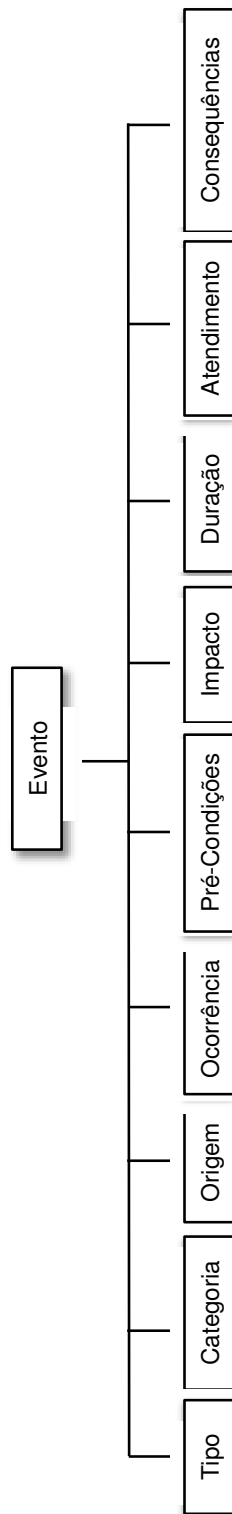


Figura 5.11: Visão de alto nível da taxonomia de eventos.

Origem: No contexto de PMAOE, todo evento é ocasionado por algum agente humano, fenômeno da natureza, ou decorre de defeito em equipamento (figura 5.12). Identificar qual a origem de um evento é importante na elaboração do conhecimento dos lugares onde eventos podem acontecer com maior probabilidade e a existência de eventos causadores.

- **Origem humana:** pode ser (i) *intencional*, quando o agente causa o evento de forma deliberada, como, por exemplo, no caso de um assalto, atentado ou vandalismo, ou (ii) *acidental*, que ocorre quando o agente causa o evento, mas sem intenção, geralmente devido à inabilidade, desatenção ou imprudência durante a realização de determinadas tarefas. Como exemplo, podemos citar colisão de automóveis, afogamento, e atropelamento.
- **Origem natural:** ocorre devido a fenômenos da natureza, sobre os quais o ser humano não tem uma interferência direta. Exemplos desse tipo de evento são os terremotos, enchentes, furacão e tsunamis.
- **Origem em equipamentos:** ocorre devido à falta de manutenção ou à forma incorreta de se utilizá-los. Exemplos são explosão de caldeira, vazamento de radioatividade em usina nuclear, “apagão” (*blackout*) ou pode ser devido a eventos imprevistos, como enchentes, sobrecarga na rede elétrica etc.

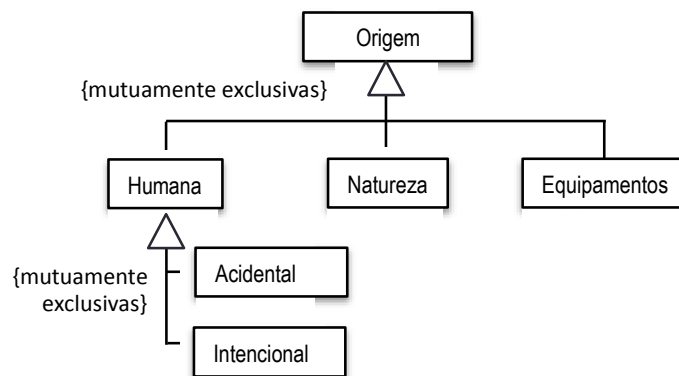


Figura 5.12: Representação da origem de um evento.

Ocorrência: conforme exposto na seção 5.2, eventos tem uma referência espaço-temporal (Lewis; 1973; Davidson, 1980; Kim, 1993; Scheneider, 2010; Casati e Varzi, 2010). Dessa forma, essa dimensão é composta por dois elementos: *momento* e *localização* (figura 5.13) O momento representa o instante inicial da

ocorrência de uma instância de evento. A localização representa o vértice onde o evento ocorre.

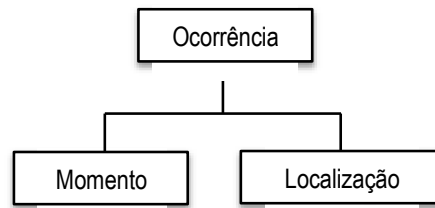


Figura 5.13: Representação da ocorrência de um evento.

Pré-Condições: conforme exposto na propriedade 4 da subseção 5.2.6, para que um evento ocorra, é preciso que um ou mais eventos causais ocorram ao mesmo tempo que determinadas propriedades assumem valores específicos. Dessa forma, conforme a figura 5.14 a dimensão “pré-condições” possui dois atributos: *evento causal* e *estado*. O primeiro representa o conjunto de eventos que pode desencadear o evento principal e o segundo representa o conjunto de propriedades que descreve o estado do vértice que contribui para a ocorrência do evento principal.

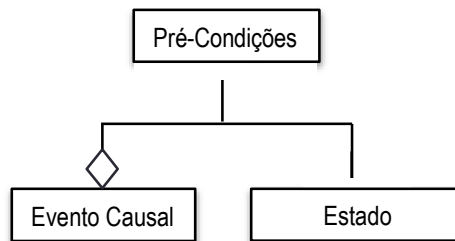


Figura 5.14: Representação de pré-condições de um evento.

Impacto: representa o quanto de prejuízo um determinado evento pode causar, tanto do ponto de vista *material* (por exemplo, bens materiais, infraestrutura etc.) como *imaterial* (vidas humanas ou não), e a abrangência (figura 5.15). Sendo assim, essa dimensão descreve a intensidade de cada evento em termos de sua capacidade de causar prejuízos ao ambiente. Ela é composta por três dimensões:

- **Prejuízos:** representam as consequências de cada evento. Podem ser de dois tipos: *tangíveis* ou *não-tangíveis*. No primeiro caso, levamos em consideração a destruição total ou parcial de bens materiais e/ou infraestrutura cujo valor pode ser medido objetivamente. No outro caso, consideramos danos a bens sobre os quais é difícil atribuir um valor

objetivo, tais como vidas humanas, obras de arte, patrimônio histórico etc. O conhecimento sobre os prejuízos será importante tanto na definição das métricas como no projeto dos agentes, mais especificamente no modelo de deliberação.

- **Abrangência:** corresponde a dois aspectos diferentes: (i) *abrangência geográfica* que representa o tamanho da área na qual o evento consegue exercer influência direta ou indireta, podendo ser desde o local imediato do evento, como por exemplo, no caso de um ato de assalto, até grande áreas, afetando um continente inteiro, como foi o caso do acidente da Usina Nuclear de Chernobyl (INSAG, 1992), e (ii) *Abrangência Temporal* que representa a duração do evento, a qual pode ter um valor desprezível tal como uma explosão até um incêndio que pode durar semanas.
- **Nocividade:** diz respeito ao potencial de dano, que representa a periculosidade de um evento, principalmente para a vida humana. Por exemplo, um incêndio causando a morte de dezenas ou centenas de pessoas, além de animais e da vegetação, não é mais letal do que o vazamento de energia nuclear, pois esse pode causar problemas tanto nas pessoas afetadas diretamente como nas pessoas localizadas em regiões distantes, através do ar, rios e consumo de alimentos, causando alterações genéticas e doenças incuráveis por décadas seguidas.

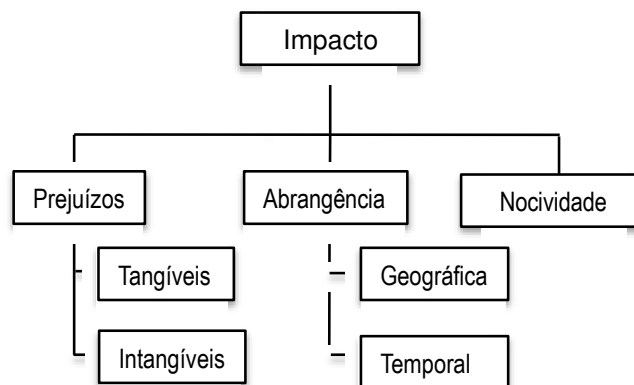


Figura 5.15: Representação de impacto de um evento.

Atendimento: em função das características de cada tipo de evento, é preciso que a visita seja realizada em um momento apropriado. Dentro dessa perspectiva, eventos podem ser *evitáveis*, *tratáveis* e *manipuláveis* (figura 5.16). Por exemplo, no caso de “afogamento”, o agente deve chegar o quanto mais cedo possível **durante** o evento. Supondo que a região em que a pessoa se encontra seja segura,

podemos supor que é praticamente impossível evitar, por diversas razões (p. ex. desmaio, infarto, queda etc.) que o evento se inicie. No entanto, é fundamental que o agente chegue o mais cedo possível a fim de minimizar os danos e/ou impedir que o evento se conclua. Caso o agente chegue após a ocorrência do evento ter concluído, isso será considerado um fracasso. Por outro lado, no caso de “acidente nuclear” ou de “colapso de caldeira”, é possível monitorar o estado dos equipamentos por meio de visitas frequentes e observação do sistema de resfriamento, e assim poder evitá-los. Isso é possível porque há meios pelos quais pode se prever com certa segurança se, dados os eventos causais se confirmem, o evento principal ocorrerá. A situação ideal seria aquela em que todos eventos fossem evitáveis. No entanto, devido à percepção dos agentes ser limitada e à quantidade de agentes não cobrir todas as regiões do grafo, é certo que a ocorrência de determinados eventos não será percebida a tempo de ser evitada ou interrompida. Por exemplo, no caso de “assalto” ou “colisão de automóveis” é praticamente impossível identificar quando um evento causal está ocorrendo de tal forma que um evento principal seja desencadeado. No caso do “assalto” não há como o agente identificar que o evento está para ocorrer e evitá-lo a não ser pela verificação de porte de arma, comportamento suspeito, intenção etc., o que inviabiliza qualquer esquema de patrulhamento baseado em agentes.

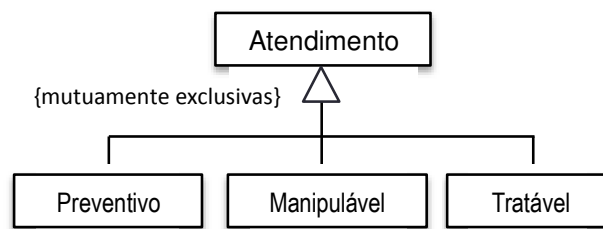


Figura 5.16: Representação da reação a um evento.

Consequências: cada *evento principal* ocorrido terá dois atributos: *propagabilidade* e *evento* (figura 5.17). O primeiro informa se o evento é capaz de se propagar para outros vértices, como por exemplo, incêndio e acidente nuclear, ou é isolado, como por exemplo, afogamento e colisão de automóveis. O segundo informa o conjunto de identificadores dos *eventos consequências*. Se o evento em questão for do tipo consequência, esse atributo deve ser vazio.

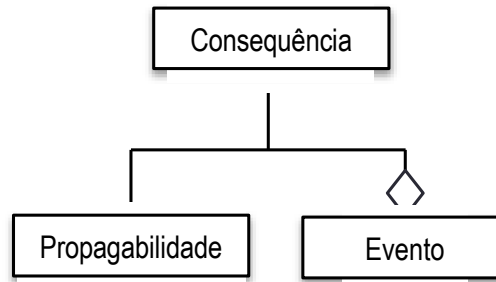


Figura 5.17: Representação da consequência a um evento.

5.3.6 Comportamento dos Eventos

A habilidade do agente em realizar patrulhamento com sucesso está na sua capacidade de conseguir identificar a ocorrência de eventos. É importante, portanto, que o agente disponha de conhecimento explícito do mundo, tanto sobre as regiões de interesse como sobre o que ocorre nelas. Além disso, sua percepção deve permitir obter informações que aumentem as chances de prever e localizar onde está ocorrendo um evento.

A probabilidade de ocorrência de cada evento pode variar ao longo do tempo de acordo com o estado corrente do mundo. Dessa forma, as prioridades de visita e o percurso a ser realizado pelos agentes irão mudar ao longo do tempo a fim de refletir essa mudança. Por exemplo, é natural supor que a probabilidade de ocorrência do evento “afogamento” aumente durante o período de 8 horas às 14 horas nos finais de semana e diminua da segunda-feira à sexta-feira no mesmo período. Essa abordagem é particularmente mais útil se compararmos com cenários estáticos como os apresentados em Elmaliach e colegas (2007), nos quais o agente sempre visitará a mesma sequência de vértices, pois o mundo é estático.

Diante do exposto e de acordo com o que foi estudado nas seções anteriores temos as seguintes propriedades:

- A intensidade de um evento varia de um tipo para outro e entre instâncias de um mesmo tipo, de acordo com uma série de fatores exógenos e que o antecedem (isso será explicado no capítulo 6);
- Um evento se divide em três partes: um intervalo de tempo antes de seu início (quando ocorre o *evento causal*), um durante (que é o *evento principal*) e outro depois (quando ocorre o *evento consequência*). Por exemplo, no caso do “colapso de caldeira” o evento causal é a

temperatura se elevar e alcançar 280° C, o evento principal é a “explosão da caldeira” e o evento consequência é “incêndio das instalações”;

- O agente sempre identifica a ocorrência de um evento quando ele visita um vértice onde está havendo uma ocorrência. Isso significa dizer que não há ruído nos sensores e que a probabilidade de identificação é igual a 100% quando o agente se encontra no mesmo vértice em que ocorre um evento;
- O agente identifica um evento no mesmo instante em que chega ao vértice: o agente tem visão completa de todo o vértice.
- Um *evento principal* ocorre imediatamente após um *evento causal*. O mesmo raciocínio vale para o *evento consequência*.

De acordo com os eventos apresentados anteriormente, classificamos cada um de acordo com a possibilidade de se interrompê-los ou evitá-los.

Quadro 5.1: Classificação dos eventos de acordo com o momento adequado de visita.

	Evitável	Manipulável	Tratável
Atentado Terrorista			●
Acidente Nuclear	●	●	●
Afogamento		●	
Assalto		●	
Colapso de Caldeira	●	●	●
Incêndio		●	
Vandalismo		●	
Colisão de Automóveis			●
Acúmulo de lixo			●

De acordo com essa classificação nós podemos então definir um conjunto de métricas que permita avaliar o desempenho dos agentes observando seu comportamento diante da possibilidade de ocorrência dos eventos de acordo com essas restrições. No próximo capítulo nós iremos apresentar esse conjunto de métricas.

5.4 Conclusões sobre o Capítulo

O objetivo desse capítulo foi apresentar a etapa de Modelagem da Classe de Problema. Através dela pudemos realizar uma organização conceitual de e apresentar uma nova perspectiva para o problema da patrulha. Nós adotamos o termo PMAOE (*Event Oriented MultiAgent Patrolling*), conforme já existente na literatura (Agmon, 2010), devido ao fato que os agentes possuem um conhecimento explícito acerca da natureza

dos eventos. Esse conhecimento descreve, dentre outras coisas, o tipo dos eventos, causalidade, consequência, impacto e sua relação com determinadas pontos de interesse do grafo.

A fim de tornar o benchmark coerente com a problemática que envolve o conceito de eventos, identificamos de acordo com a Metafísica quais características são pertinentes à nossa proposta e propusemos uma definição para PMAOE de acordo com a Teoria dos Eventos e com a tarefa da patrulha. Dessa forma, consideramos que nossa definição possui o rigor necessário para ser utilizado na compreensão e desenvolvimento de soluções para o problema.

Uma vantagem que vislumbramos nessa abordagem é que eventos são caracterizados de tal forma que os agentes podem tomar decisões de acordo com o estado corrente do ambiente. Assim, eventos podem ser evitados, manipulados ou tratados, evitando ou mitigando seus efeitos. Dessa maneira, PMAOE pode ser de grande utilidade para o estudo de problemas relacionados com segurança, tais como gestão e recuperação de desastres, segurança preventiva, monitoração de recursos, localização de pessoas perdidas etc. Com o propósito de demonstrar o potencial de nossa proposta em prover cenários úteis e realistas, ilustramos um exemplo em que agentes atuando sob a maneira tradicional e sob a nossa perspectiva lidam com eventos capazes de causar danos físicos e imateriais.

Este capítulo apresentou três contribuições importantes: a primeira foi a organização conceitual da pesquisa de através de uma taxonomia; a segunda foi a apresentação de uma nova definição para o problema da patrulha, a qual abrange qualquer tipo de trabalho de , não tem qualquer viés em relação à estratégia a ser adotada e é independente de métrica; o desenvolvimento de um novo tipo de patrulha na qual os agentes possuem conhecimento explícito sobre o mundo em que estão situados; a conceituação do termo evento levando em consideração as especificidades do problema da patrulha e da Teoria dos Eventos, e a criação de uma taxonomia para descrever os diversos tipos de eventos.

Capítulo 6

Construção dos Cenários

O objetivo deste capítulo é apresentar a fase de construção de cenários do benchmark OEMAP de acordo com PROBENCH. Em função de sua importância para o sucesso do benchmark, faz-se necessário que esse conjunto de diretrizes conduza os desenvolvedores em sua obtenção. Apesar de que ela consiste na instanciamento das taxonomias apresentadas no capítulo 5, é fundamental que critérios sejam adotados na sua elaboração. Adicionalmente, uma quantidade excessiva de instâncias pode ser obtida caso sejam empregadas combinações de valores sem nenhum direcionamento. Para isso, ao final do capítulo, discutimos a importância e aplicabilidade dos cenários no contexto do problema da patrulha¹.

6.1 Descrição dos Cenários de Interesse

Esta seção discute cenários que podem ser utilizados na condução dos experimentos em PMAOE. Seu objetivo é identificar quais aspectos da classe de problema e, consequentemente, quais atributos influenciam o desempenho da solução, para assim poder orientar os desenvolvedores na identificação de quais são os cenários são mais apropriados para a realização dos experimentos.

6.1.1 Grafo

Conforme foi demonstrado no trabalho de Machado e colegas (2002) no contexto de MATP, o tipo de grafo influencia o desempenho dos agentes. Isso nos leva a acreditar que outros estudos devem ser realizados no sentido de investigar como outras características do grafo podem influenciar a habilidade dos agentes em identificar eventos em PMAOE.

¹ A atividade denominada *Armazenamento de Valor* foi omitida neste capítulo por se tratar de uma atividade temporária e incremental. Os valores obtidos por meio dela serão mostrados na atividade seguinte, denominada *Instanciamento de cenários*.

Quanto ao Tipo: nesse tipo de cenário estamos interessados em analisar o desempenho dos agentes nos diferentes tipos de topologias, independente das demais características do cenário (sociedade, eventos e agentes). Cada instância deverá considerar um tipo de topologia diferente conforme apresentado na seção 4.2.2. Nesse cenário podemos observar o seguinte: os grafos sugeridos são aqueles já estudados na literatura (Machado et al., 2002) e a informação que varia é unicamente o tipo de grafo. Quanto às demais, são adotadas as situações mais simples possível: grafo estático, sem prioridade, duração da visita constante e uniforme, custo constante do arco e arco unidirecional.

Quanto à Dinamicidade do Vértice: outro tipo de cenário interessante para o estudo de agentes é quanto à dinamicidade do vértice. Esse é o tipo de grafo que modela situações dinâmicas do mundo tais como aqueles casos em que um banco ou usina encerra suas atividades, ou um setor de uma fábrica tem sua porta trancada, desabilitando seu acesso. Essa situação tanto pode ser definitiva como temporária. No primeiro caso, o agente precisa atualizar sua base de conhecimento sempre que identificar alguma mudança em relação a um ponto de interesse, fazendo com que ele redefina prioridades e, conseqüentemente, adote novas rotas. Uma vez que a dinamicidade do vértice está diretamente relacionada com a topologia pode ser importante também analisar qual o efeito da dinamicidade no desempenho dos agentes em cada uma das topologias. Por exemplo, em grafos minimamente conectados, tais como corredor ou circular, a indisponibilidade de um determinado grafo levará aos agentes a redefinirem totalmente suas rotas. Por outro lado, em grafos altamente conectados, tais como grafos irregulares ou matrizes, o agente poderá manter a mesma rota.

Quanto à Prioridade do Vértice: a prioridade de um vértice tem grande importância no projeto de agentes para o problema da patrulha visto que o objetivo é evitar ou mitigar incidentes de qualquer natureza. Nesse caso, alguns pesquisadores podem querer projetar agentes que irão lidar apenas com um tipo de evento (e.g. acidente nuclear ou afogamento), enquanto outros podem querer projetar ambientes consistindo de diversos tipos de eventos, cada um com um nível de prioridade. Nesse segundo caso, é importante observar que a prioridade pode variar de acordo com várias características do mundo. Por exemplo, conforme já explicado na seção 5.9, a prioridade de uma praia varia de acordo com horário, condições climáticas e aspectos sociais (como exemplo feriados e fins de semana).

Quanto à Duração da Visita: na PMAOE a duração da visita descreve a quantidade de tempo que o agente leva para detectar um evento quando chega a um determinado

ponto de interesse da região patrulhada. Esse tipo de situação pode ocorrer, por exemplo, em lugares de difícil visibilidade ou acesso. Quando a duração é uniforme (mesma valor para todos os vértices) esse atributo não interfere no desempenho dos agentes. No entanto, em determinados domínios pode ser interessante modelar o tempo de visita como sendo variável a fim de demonstrar a complexidade de identificar determinados eventos. Por exemplo, a identificação de um afogamento pode ser imediata enquanto a de um acidente nuclear pode demandar a análise da temperatura do sistema de resfriamento, dos níveis de radioatividade no ambiente etc.

Quanto ao Custo do Arco: Similarmente ao caso anterior, o custo do arco tem uma influência direta no desempenho dos agentes. O custo do arco modela um conjunto de situações do mundo real com as quais as patrulhas podem se deparar: estradas com pedágio, ladeiras etc. Dessa forma, o agente pode se deparar com a situação em que o caminho de um vértice a outro é menor, porém demandando maior gasto financeiro ou de energia, enquanto outro é maior, porém com menor gasto financeiro ou de energia.

Quanto ao Sentido do Arco: Em alguns contextos, como por exemplo, o patrulhamento de cidades, onde os agentes modelam viaturas policiais ou dos bombeiros, os caminhos podem ter sentidos (ou direcionalidade), o que pode implicar em uma restrição na forma como os agentes escolhem os vértices que serão visitados. O agente pode, por exemplo, se encontrar em um vértice diretamente conectado a outro que ele deve visitar, mas o arco que os ligam não permite ser atravessado como entrada, e sim como saída.

6.1.2 Quanto à Sociedade

O estudo de agentes em cenários baseados na sociedade é essencial para a análise sobre como habilidades relacionadas à cooperação, negociação e adaptabilidade podem influenciar na identificação de eventos.

Quanto à Homogeneidade: Nesse cenário o pesquisador deve analisar os efeitos de se ter uma abordagem baseada em agentes com as mesmas habilidades (homogênea) e outra, na qual os agentes têm diferentes habilidades (heterogênea). No trabalho de Almeida e colegas (2003) é apresentada uma abordagem denominada de HPCC (*Heuristic Path Finder Cognitive Coordinator*) na qual um agente coordena a sociedade a fim de identificar quais vértices serão visitados pelos demais agentes.

Quanto à Dinamicidade: Nesse tipo de cenário será avaliado se a dinamicidade da sociedade interfere na habilidade dos agentes em detectar eventos. A princípio vislumbramos duas situações em que tipo de cenário é interessante: a primeira é quando o objetivo do estudo diz respeito às características sociais do agente: cooperação, negociação e comunicação. O aumento ou diminuição da sociedade pode ter reflexos na forma como os mecanismos subjacentes a essas características deve ser ajustado ao longo do tempo, de acordo com a quantidade de agentes na sociedade. Essa restrição pode permitir, por exemplo, o estudo da adaptabilidade do agente perante uma sociedade dinâmica. A segunda é quando a região é particionada entre os agentes, de tal forma que cada um deles se responsabiliza por cada sub-região. O aumento ou diminuição da sociedade implicará que seja realizada uma nova redistribuição dos vértices a cada aumento ou diminuição da sociedade. Nesse caso estudos podem considerar de que forma o agente irá ajustar sua base de conhecimento a cada mudança do tamanho da sociedade;

Comunicação: Nesse tipo de cenário será avaliada de que forma a habilidade de se comunicar contribui para a identificação de eventos. Podem ser analisado se agentes sem comunicação são mais eficientes do que agentes que se comunicam, e nesse caso qual o modelo mais eficiente: troca de mensagens, quadro negro ou sinais (conforme apresentado no capítulo 4). Uma situação que vislumbramos é a de agentes que compartilham informações a respeito de onde há mais chances de ocorrerem eventos e qual a prioridade de tal forma que quem estiver mais próximo, visite o vértice.

6.1.3 Quanto aos Eventos

Conforme visto na subseção 5.3.5 Taxonomia de Eventos, eventos tem diversas características, as quais influenciam diretamente a forma como o patrulhamento deve ser realizado. No entanto, nesse trabalho iremos analisar unicamente a capacidade dos agentes em identificar eventos, sem levar em consideração características mais avançadas tais como habilidade de detectar múltiplas ocorrências simultâneas, eventos propagáveis etc.

Tipo: o tipo de evento é importante para a realização de diversos estudos acerca do comportamento dos agentes projetados. Por exemplo, para saber se eles são mais eficientes na prevenção, tratamento ou manipulação dos eventos; se os agentes estão detectando mais eventos de menor impacto ou menos de maior impacto; se

os agentes estão satisfazendo a necessidade de patrulhamento (ou seja, o impacto dos eventos não detectados se encontra dentro de um percentual aceitável).

6.1.4 Quanto aos Agentes

Apesar de que o projeto dos agentes é uma atividade independente da definição dos cenários, algumas restrições são impostas pelo problema, tais como se os agentes dispõem de informações a priori e qual é o alcance da percepção.

Informação a Priori: os agentes podem dispor ou não de informação a priori. Em domínios tais como o de exploração de ambientes desconhecidos os agentes devem inicialmente, procurar fazer uma visita de reconhecimento a fim de descobrir a estrutura do ambiente antes de começar a patrulha. Quando o agente dispõe de informações dessas informações como, por exemplo, no caso de uma guerra, em que os soldados dispõem de mapas essa etapa inicial é desnecessária. As informações a priori disponíveis aos agentes são a *sociedade* e o *grafo*. No primeiro caso o agente pode ser importante ao agente saber a quantidade de agentes na sociedade por vários motivos: estratégias de coordenação, divisão da região patrulhada, estratégias de negociação etc. No outro caso, o agente precisa conhecer o grafo que será patrulhado, tornando desnecessária a etapa de reconhecimento do terreno.

Quanto à Percepção: a percepção de um agente se baseia em dois tipos de elementos conforme visto na subseção 5.3.5 Taxonomia de Eventos: o *objeto que pode ser percebido* e *alcance da percepção*. No primeiro caso, o agente pode ser capaz de perceber a cada ciclo uma série de características do mundo: *grafo*, *sociedade*, *custo do arco*, *evento* e *tempo*. No segundo caso, o alcance diz respeito à distância na qual os agentes conseguem perceber os eventos. Para fins de simplicidade, essa distância pode ser dada em termos de vértices, onde $p = 1$ permite que o agente perceba os vértices diretamente conectados, $p = 2$ os vértices diretamente conectados e os que estão diretamente ligados a estes, e assim por diante.

6.1.5 Quanto à Dinamicidade do Mundo

Contrastando com os cenários apresentados até então, onde há um maior controle das variáveis primárias, aqui propomos um tipo de cenário onde várias propriedades podem variar ao mesmo tempo. O agente deve ser capaz de lidar com um mundo onde há um alto grau de imprevisibilidade o que torna o problema bem mais complexo do que os

casos anteriores. Perguntas como as seguintes poder ser respondidas pelos agentes a cada ciclo de percepção:

1. O grafo continua o mesmo?
2. Algum vértice deixou de existir?
3. Algum arco está indisponível?
4. O custo dos arcos ainda é o mesmo?
5. Algum evento está ocorrendo no vértice corrente?
6. Algum evento pode ocorrer nos vértices dentro do alcance de percepção?
7. A quantidade de agentes é a mesma?

Para isso sugerimos que os vértices sejam dinâmicos, que a duração da visita seja variável e a sociedade seja dinâmica, com vários tipos de eventos,

6.2 Identificação dos Atributos Primários e Secundários

Tendo analisado os cenários de interesse, os seguintes elementos foram identificados como sendo **aspectos de desempenho** para PMAOE: *agente, sociedade, grafo e evento*. A motivação para essa escolha se deu pelo fato de que cada aspecto possui atributos que influenciam diretamente no comportamento do SMA a ser estudado por meio de PMAOE. Conforme, vimos na seção 3.5 Construção dos , a importância e, conseqüentemente, escolha, dos atributos depende do tipo de estudo que os pesquisadores desejam realizar.

Os seguintes atributos foram definidos como sendo primários:

- **Grafo:** tipo, vértice (dinamicidade, prioridade, duração da visita), arco (custo, direção, dinamicidade) e tamanho;
- **Sociedade:** organização, dinamicidade, comunicação e tamanho;
- **Agente:** objeto da percepção;
- **Evento:** tipo;

Os seguintes atributos da classe de problema foram definidos como sendo secundários:

- **Sociedade:** organização e dinamicidade;
- **Grafo:** disponibilidade, prioridade, duração da visita, custo, direção, disponibilidade e tamanho;
- **Agente** percepção a priori da sociedade e do grafo;
- **Evento:** todos os atributos apresentados em 5.3.5 com exceção do atributo *tipo*.

6.3 Identificação de Restrições

Nessa seção nós descrevemos a aplicação de PROBENCH na identificação das restrições de cada atributo das taxonomias de PMAOE. Cada restrição permite identificar de que forma cada atributo será definido. No quadro 6.1 temos um sumário do resultado dessa atividade.

No caso de **grafo** os atributos são definidos por meio de três restrições conforme apresentado na subseção 3. Inicialmente, *duração da visita*, *custo* e *direção* terão *valor default*. A razão para isso é que em grande parte da literatura de são usados os mesmos valores, o que contribui para a padronização do problema e, conseqüente, melhoria da comparatividade das soluções desenvolvidas. Em seguida, os valores dos atributos *prioridade*, *dinamicidade* do arco e *dinamicidade* do vértice são definidos por meio de *decisão de projeto*. Uma vez que iremos lidar com cenários simples nesse estágio da pesquisa, decidimos não abordar características mais complexas do problema. No que diz respeito à restrição *imposição do problema*, temos os atributos *tipo* e *tamanho*. Esses valores serão obtidos juntos à declaração do problema ou à definição do mundo em que os agentes da solução estarão inseridos.

No caso de **sociedade**, a restrição para os atributos *organização* e *dinamicidade* é dada por *decisão de projeto*. Conforme mencionado no parágrafo anterior, decidimos que nesse estágio inicial da pesquisa, esses atributos terão esses valores tão simples quanto possível. Os atributos *comunicação* e *tamanho* dependem do problema em questão. Sendo assim, seus valores serão definidos de acordo com a imposição do problema.

Em relação a **evento**, todos os atributos serão definidos por meio de *decisão de projeto*. Esse atributo, em especial, acrescenta inúmeras possibilidades ao estudo da patrulha. Uma vez que nosso objetivo é verificar a viabilidade de PROBENCH como benchmark, evento será estudado em detalhes nos trabalhos futuros. Portanto, a definição de seus atributos será restrita como *decisão de projeto*.

Similarmente, iremos definir todos os atributos de **agente** de acordo com a restrição de *decisão de projeto* pelas mesmas razões relacionadas a *evento*.

Quadro 6.1: Restrições dos atributos de PMAOE.

Restrição			
Dimensão	Valor default	Decisão de projeto	Imposição do problema
Grafo	duração da visita, custo, direção.	prioridade, dinamicidade do vértice e do arco.	tipo, tamanho
Sociedade		organização, dinamicidade	comunicação, tamanho
Evento		todos os atributos	
Agente		todos os atributos	

6.6 Pesquisa Descrição de Problema

Conforme visto na seção anterior, os valores de determinados atributos foram definidos através de pesquisa na *definição do problema*. Adicionalmente, analisamos a definição de PMAOE apresentada na seção 5.2.2, adotamos os seguintes valores:

- **Grafo:** prioridade (*verdadeiro, falso*), dinamicidade (*verdadeiro, falso*) do vértice e do arco;
- **Sociedade:** organização (*heterogênea, homogênea*) e dinamicidade (*verdadeiro, falso*);

6.5 Pesquisa Literatura

Uma vez que nosso grupo de pesquisa já vem trabalhando no problema da patrulha há vários anos, já dispunhamos de vários *valores default* conforme apresentado na seção 6.3. Dessa forma adotamos os seguintes valores de **grafo** foram obtidos:

- **duração da visita:** *um ciclo*. Esse valor é usado para qualquer visita, independente do vértice e do evento;

- **custo:** uma vez que iremos adotar os grafos existentes na literatura, mantivemos os custos dos arcos já utilizados em trabalhos como os de Machado e colegas (2002) e Almeida e colegas (2004).
- **direção:** *unidirecional*. O valor adotado pelos trabalhos desenvolvidos em se refere a arcos unidirecionais.

6.7 Criação de Valores

Os valores dos seguintes atributos foram criados de acordo com um conjunto de *decisões de projeto* uma vez que não dispunhamos de trabalhos existentes que abordassem explicitamente esses conceitos:

- **Agente:** informação a priori da sociedade: *sim* (esse valor foi escolhido porque consideramos que, ao iniciar o experimento tendo todas as informações básicas, os pesquisadores podem focar na habilidade de identificar eventos); informação a priori do grafo: *sim*; percepção do grafo: *sim*; percepção da sociedade: *sim*; percepção do evento: *sim*; percepção do tempo: *sim*; alcance: 1 (decidimos escolher esse valor a fim de limitar o alcance do agente apenas aos vértices diretamente alcançáveis); confiabilidade: *sim* (esse valor significa que a informação percebida pelo agente é totalmente confiável. Dessa forma, decidimos não incluir sensores com ruídos nessa etapa inicial da pesquisa em PMAOE, deixando-os para trabalhos futuros).
- **Evento:** todos os atributos;

6.8 Instanciação de Cenários

Os *datasets* criados através de PROBENCH devem ser obtidos por meio das sub-atividades apresentadas na Figura 3.5: Instanciação de cenários. Uma vez tendo definido o objetivo do cenário a ser criado através da atividade *descrição dos cenários de interesse* e estando de posse dos valores dos atributos primários e secundários, foram criadas as instâncias dos cenários .

A título de exemplo, nos quadros 6.2, 6.3 e 6.4 são apresentados os valores adotados para cada atributo de PMAOE que define os cenários. No capítulo Anexo A um conjunto de *datasets* para PMAOE é são apresentado.

Quadro 6.2: valores de grafo.

Grafo		
Tipo	Mapa A, Mapa B, Circular, Corredor	
Vértice	Disponibilidade	Sim
	Prioridade	Não
	Duração da visita	Irrelevante
Arco	Custo	Heterogêneo
	Direção	Unidirecional
	Disponibilidade	Sim
Tamanho	Serão adotados os tamanhos previstos na literatura	

Quadro 6.3: valores de sociedade.

Sociedade	
Organização	Homogênea
Dinamicidade	Estática
Comunicação	Troca de mensagens
Tamanho	5, 10, 15, 20, 25

Quadro 6.4: valores de agente.

Agente			
Informação a Priori	Sociedade	Sim	
	Grafo	Sim	
Percepção	Objeto	Grafo	Sim
		Sociedade	Sim
		Custo do arco	Sim
		Evento	Sim
		Tempo	Sim
	Alcance	Constante (1)	
	Confiabilidade	Sim	

6.10 Conclusões sobre o Capítulo

Neste capítulo apresentamos de que forma PROBENCH pode ser utilizado na criação de um conjunto de cenários no contexto de OEMAP. É preciso ressaltar que cenários consistem em um dos artefatos mais importantes de um benchmark e que, portanto, devem ser criados dentro de um processo sistemático a fim de minimizar as chances de erro. Nesse trabalho buscamos, além de demonstrar a aplicabilidade de PROBENCH, suprir uma importante lacuna no contexto de SMA, mais especificamente no problema da patrulha: dispor um conjunto de cenários de testes que possa ser compartilhado por toda a comunidade de PMA. Apesar de ser um conjunto amplo, ele não é completo uma vez que decidimos por adotar cenários simples nesse estágio da pesquisa.

O uso de um processo foi fundamental na condução dessa etapa visto que através dele pudemos identificar quais artefatos deveriam ser produzidos e como a atividade deveria ser executada detalhadamente. Como resultado, acreditamos que PROBENCH se mostra um meio eficiente na construção de cenários para a realização de experimentos por meio de benchmarks.

Capítulo 7

Definição das Métricas

De acordo com PROBENCH, após concluída a etapa de construção dos cenários, devemos realizar a definição das métricas. Conforme vimos na seção 4.3.3, as métricas adotadas pela grande maioria dos trabalhos em se baseiam no conceito de *tempo*, como por exemplo, *ociosidade* e *frequência*. A principal razão para a adoção desse tipo de métrica vem do fato que nos tipos de patrulhas existentes os agentes não possuem conhecimento explícito acerca dos eventos e, por consequência, seu critério-objetivo se restringe a minimizar o intervalo de tempo entre visitas consecutivas ou a maximizar a frequência de visitas a cada vértice. Nesse trabalho os agentes possuem conhecimento explícito sobre os eventos que ocorrem no mundo em que estão inseridos. Em função disso, novos critérios e, consequentemente, novas métricas devem ser criadas.

O objetivo desse capítulo é demonstrar como PROBENCH pode auxiliar pesquisadores na definição de métricas durante a criação de benchmarks para SMA. Sua aplicabilidade na condução da criação de PMAOE permitiu identificar, dentre outras coisas, quais aspectos do desempenho de um SMA devem ser avaliados a fim de compará-las bem como de compreender adequadamente o comportamento de cada solução proposta.

7.1 Identificação dos Serviços dos Agentes

O serviço oferecido pelos agentes na PMAOE é identificar a ocorrência de eventos. Conforme visto no capítulo 5, os agentes devem realizar essa tarefa se deslocando de um vértice a outro de acordo com seus tipos e as características dos eventos que podem ocorrer em cada um. Conforme visto na seção 5.3.6, dependendo do tipo de evento, agentes deverão evitar sua ocorrência, enquanto em outros devem chegar o mais cedo após o início e nos demais, devem chegar o mais cedo após seu término. Desta forma, as métricas para medir o desempenho em PMAOE devem levar em consideração cada uma dessas características do comportamento dos eventos.

Nas próximas seções iremos descrever como as métricas de OEMAP foram definidas por meio de PROBENCH.

7.2 Identificação dos Requisitos de Serviço

A avaliação de desempenho de SMA em PMAOE deve levar em consideração o *sucesso* do agente em realizar o patrulhamento. No escopo desse trabalho, *sucesso* é definido informalmente como sendo o resultado de uma visita a um vértice, a qual deve ocorrer no mesmo instante de tempo em que um evento esteja ocorrendo. Assim, o agente poderá identificá-lo e, conseqüentemente, interrompê-lo. Dentro dessa perspectiva, as métricas devem atender aos requisitos abaixo:

- **Eficiência:** representa a necessidade de avaliar a capacidade do agente em identificar a maior quantidade possível de eventos. Informações relacionadas a cada evento tais como instante adequado de identificação, atraso de identificação e fator de impacto não serão consideradas pelos agentes. Todos os eventos são tratados da mesma maneira. Apesar de que essas métricas são bastante simples diante do conjunto de avaliações e estudos que podem ser feitos em PMAOE, elas fornecem uma visão bastante útil na realização de experimentos. Esse requisito é interessante, por exemplo, na realização de pesquisas onde o cenário possui apenas um tipo de evento, uma vez que os eventos não dispõem informações acerca da nocividade e, conseqüentemente, da importância. Sendo assim, o usuário de PMAOE deve ser capaz de responder a seguinte pergunta: *Os agentes conseguem identificar uma quantidade satisfatória de eventos?* Pelo termo “satisfatória” queremos dizer que a solução possui um desempenho igual ou acima de um valor esperado, ou melhor do que as concorrentes.
- **Responsividade:** A responsividade é um determinante da qualidade de serviços, sendo entendida como a prontidão no atendimento dos clientes ou como o tempo que estes devem esperar por resposta ou atenção. Em outras palavras, responsividade representa a necessidade de avaliar a habilidade do agente em identificar a ocorrência de cada evento o mais cedo possível após seu início (Soratto e Varvakis, 2005). Esse requisito é importante, sobretudo, naqueles casos em que há difícil previsibilidade mas há possibilidade de interrompê-lo, tais como aqueles casos em que há gestão de desastres, afogamentos, colis. O usuário de PMAOE deve ser capaz, portanto, a seguinte pergunta: *Os agentes conseguem identificar rapidamente os eventos?*
- **Mitigabilidade:** representa a necessidade de avaliar a habilidade dos agentes em mitigar a maior quantidade de possível de prejuízos. Para isso os agentes devem observar, sobretudo, o *fator de impacto* de cada evento.

Assim, o usuário de PMAOE deve ser capaz de responder à seguinte pergunta: *Os agentes conseguem mitigar os prejuízos de forma satisfatória?*

É preciso compreender a relação entre os requisitos supracitados e as características dos eventos a fim de que possamos propor um conjunto útil de métricas. Por exemplo, alguns eventos tais como “acidente nuclear” e “colapso de caldeira” demandam *proatividade* por parte dos agentes, enquanto “colisão de automóveis” e “afogamento” demandam *reatividade*. Essa compreensão influenciará diretamente o projeto dos agentes. A partir desses requisitos propomos os seguintes critérios-objetivos:

- Maximizar a quantidade de eventos identificados;
- Minimizar o atraso na identificação de eventos;
- Minimizar a quantidade de prejuízo decorrente dos eventos ocorridos;

Nas próximas seções iremos definir as métricas que serão utilizadas para avaliar os agentes de acordo com cada um dos critérios conforme especificado no capítulo 3.

7.3 Identificação das Métricas de acordo com Eficiência

Nessa seção iremos descrever detalhadamente um conjunto de métricas proposto para PMAOE.

7.3.1 Sucesso Baseado em Quantidade

Conforme vimos na seção 5.6, um agente deve visitar as regiões de interesse à procura de eventos e observando o estado atual do mundo. Uma vez que as probabilidades dos eventos mudam ao longo do tempo, a decisão sobre qual vértice será visitado varia entre visitas consecutivas. Dessa forma, na PMAOE a sequência dos vértices visitados muda ao longo do tempo. Consequentemente, os agentes terão que adotar uma estratégia de visitação não-uniforme. Isso é diferente de métricas tais como aquelas baseadas em ociosidade ou frequência de visita. Por exemplo, na abordagem adotada em trabalhos como o de Elmaliach e colegas (2007), é definida uma sequência fixa de visitas ao longo de toda a simulação.

Como a ociosidade é uma variável que cresce linearmente com o tempo e uniformemente em todos os vértices, é bastante natural supor que os agentes podem adotar uma abordagem estática. Como não há conhecimento explícito sobre os eventos,

o usuário poderá apenas discutir acerca da capacidade dos agentes em reduzir o valor da ociosidade a zero em cada visita, independente se ali de fato está ocorrendo um evento.

O *sucesso da visita* é definido de acordo com o resultado de cada visita, ou seja se o agente consegue ou não visitar os vértices no mesmo instante em que ocorrem eventos, independente do valor corrente da ociosidade. Por meio dele, definiremos uma função $S(v)$, onde $t(v)$ é o instante da visita ao vértice V , $t(i)$ é o instante inicial e $t(f)$ é o instante final do evento. Conforme ilustrado na figura 7.1, se o instante da visita ocorrer entre o início e o fim do evento, então consideramos que a visita foi um sucesso: $S(v) = 1$ (100%). Do contrário, não houve sucesso, ou seja $S(v) = 0$ (0%).

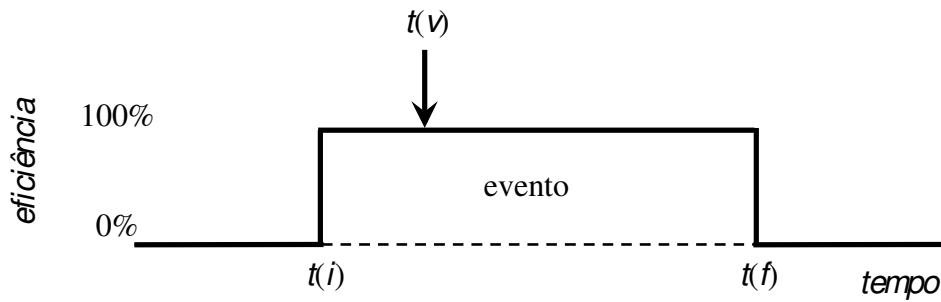


Figura 7.1: Representação de um evento do ponto de vista de sua duração e do instante de visita do agente ao vértice.

Dessa forma podemos estabelecer uma função para *sucesso* da seguinte forma:

$$S(v) = \begin{cases} 0 & \text{se evento } E_i \text{ não for identificado} \\ 1 & \text{se evento } E_i \text{ for identificado} \end{cases}$$

Por *identificar um evento* queremos dizer o ato do agente visitar um vértice V durante a ocorrência de um evento, seja ele um evento *causal*, *principal* ou *consequência*. O instante de tempo em que essa visita ocorre pode ser qualquer um durante a ocorrência do evento.

Na próxima subseção iremos apresentar as métricas criadas de acordo com o requisito de eficiência.

7.3.2 Métricas Baseadas em Eficiência

As métricas apresentadas nesta subseção tem como objetivo permitir o conhecimento acerca da habilidade dos agentes em detectar a maior quantidade possível de eventos, independente de suas características.

Inicialmente, a *quantidade total de eventos identificados* permite conhecer a quantidade total de eventos identificados no grafo G ao longo de toda a simulação, independente de seu momento adequado de identificação. Isso significa dizer que toda vez que o agente identificar um evento, ou seja quando $\mathcal{S}(v) = 1$, então o somatório será incrementado em 1 (Equação 7.1):

$$QEI(G) = \sum_{e \in E} \mathcal{S}(v)$$

Equação 7.1: Quantidade Total de Eventos Identificados.

Em alguns casos, no entanto, o pesquisador pode querer avaliar o desempenho dos agentes em identificar os eventos levando em consideração o momento ideal de identificação. Por exemplo, ele pode querer conhecer a habilidade dos agentes em identificar *eventos evitáveis* em detrimento de *eventos tratáveis*. Sendo assim, propomos três tipos de métricas de acordo com essa restrição: *quantidade total de eventos evitados*, *quantidade total de eventos manipulados* e *quantidade total de eventos tratados*.

No primeiro caso (*quantidade total de eventos evitados*) é calculada a quantidade de *eventos causais* identificados pelos agentes em todo o grafo G ao longo da simulação. Portanto, o agente deve visitar o vértice em um instante $t(v) \geq t_{E_A}(i)$ e $t(v) \leq t_{E_A}(f)$, onde E_A é um *evento causal* qualquer (Equação 7.2).

$$QEE(G) = \sum_{e \in E_A} \mathcal{S}(v)$$

Equação 7.2: Quantidade Total de Eventos Evitados.

No segundo caso (*quantidade total de eventos manipulados*) é calculada a quantidade de *eventos principais* identificados pelos agentes em todo o grafo G ao longo da simulação. Isto permite avaliar o desempenho dos agentes em relação à habilidade de identificar os eventos no momento em que eles estão ocorrendo. Portanto, o agente deve visitar o vértice em um instante $t(v) \geq t_{E_P}(i)$ e $t(v) \leq t_{E_P}(f)$, onde E_P é um *evento principal* (Equação 7.3).

$$QEM(G) = \sum_{e \in E_p} \mathcal{S}(\mathcal{V})$$

Equação 7.3: Quantidade Total de Eventos Manipulados.

No terceiro caso (*quantidade total de eventos tratados*) é calculada a quantidade de *eventos consequência* identificados pelos agentes em todo o grafo G ao longo da simulação. Isto permite avaliar o desempenho dos agentes em relação à habilidade de identificar as consequências dos eventos depois que eles venham a ocorrer. Portanto, o agente deve visitar o vértice em um instante $t(v) \geq t_{E_D}(i)$ e $t(v) \leq t_{E_D}(f)$, onde E_D é um *evento consequência* qualquer (Equação 7.4).

$$QEE(G) = \sum_{e \in E_D} \mathcal{S}(\mathcal{V})$$

Equação 7.4: Quantidade Total de Eventos Tratados.

No último caso, leva-se em consideração a quantidade de eventos que ocorrem ao longo da simulação. Conforme apresentado na equação Equação 7.5, *quantidade total relativa de eventos identificados* permite realizar um julgamento justo acerca do desempenho dos agentes uma vez que a quantidade de eventos ocorridos pode mudar entre uma solução e outra.

$$QEE(G) = \frac{\sum_{e \in E} \mathcal{S}(\mathcal{V})}{|E|}$$

Equação 7.5: Quantidade Total Relativa de Eventos Identificados.

7.3 Identificação das Métricas de acordo com a Responsividade

Conforme discutimos nos capítulos anteriores, o tempo desempenha um papel de extrema importância no projeto de soluções voltadas à monitoração e vigilância (ver subseção 5.2.6 Definição de Evento no Contexto da Patrulha). Dessa forma, é preciso que estendamos o conceito de sucesso a fim de acomodar essa necessidade e possibilitar o desenvolvimento de agentes que lidem com domínios com requisitos de temporalidade.

7.5.1 Atraso de Identificação

Em alguns tipos de experimentos é fundamental conhecer quanto tempo os agentes demoram para identificar determinados eventos logo que a ocorrência destes se inicia, e assim poder avaliar a **responsividade**. Isso é importante, sobretudo, em situações onde o agente não tem como ou não consegue evitar os eventos, como por exemplo o afogamento ou acúmulo de lixo, nos quais a demora em identificá-los e, conseqüentemente, tratá-los agrava suas conseqüências.

O **atraso** é definido, portanto, como sendo a diferença entre o instante em que o agente visita o vértice onde está ocorrendo um evento ($t(v)$) e seu instante inicial ($t(i)$), conforme ilustrado na figura 7.2.

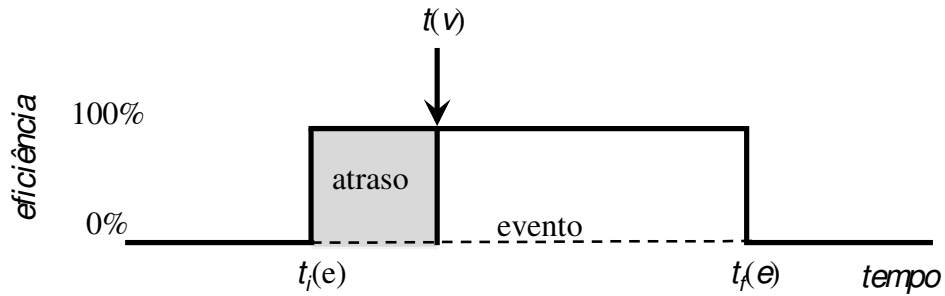


Figura 7.2: Atraso de identificação.

Para isso definimos a função $\mathcal{D}(\mathcal{V})$ como sendo o atraso que o agente leva para identificar um evento (equação 7.5). É importante ressaltar que estamos assumindo que o agente identifica o evento imediatamente após a sua chegada ao vértice.

$$\mathcal{D}(\mathcal{V}) = T(\mathcal{V}) - T(i)$$

Equação 7.6: Função atraso de identificação.

Assumindo que o tempo de simulação é discreto e baseado em ciclos, $\mathcal{D}(v)$ é a quantidade de ciclos que o agente leva para identificar a ocorrência de um evento em um determinado vértice.

A fim de permitir esse estudo no estudo da patrulha, iremos propor um conjunto de métricas baseadas no tempo. Elas estão divididas em dois grupos. O primeiro é chamado de métricas básicas, pois são basicamente métricas de contagem: *atraso mínimo de um agente*, *atraso máximo de um agente*, *atraso médio de um agente*, *atraso mínimo da sociedade*, *atraso máximo da sociedade*, *atraso médio da sociedade* e *desvio*

padrão dos atrasos. O segundo é uma interpretação do conceito de sucesso quanto à eficácia dos agentes.

7.5.2 Métricas Básicas

A métrica *Atraso Mínimo de um Agente* identifica o *menor atraso* experimentado por um agente \mathcal{A}_i durante toda a simulação. Seja $\mathcal{D}_i(\mathcal{A})$ o atraso que o agente \mathcal{A} obteve ao fazer a i -ésima identificação e Min a função que retorna o menor valor obtido entre o conjunto de atrasos obtidos, então $\mathcal{T}_{Min}^{\mathcal{A}}$ é o atraso mínimo do agente \mathcal{A} . Essa métrica pode ser obtida por meio da equação 7.6:

$$\mathcal{T}_{min}^{\mathcal{A}} = Min[\mathcal{D}_1(\mathcal{A}), \mathcal{D}_2(\mathcal{A}), \dots, \mathcal{D}_n(\mathcal{A})]$$

Equação 7.7: Atraso mínimo experimentado por um agente.

A métrica *Atraso Máximo de um Agente* permite identificar o *maior atraso* experimentado por um agente \mathcal{A}_i durante toda a simulação. Seja $\mathcal{D}_i(\mathcal{A})$ o atraso que o agente \mathcal{A} obteve ao fazer a i -ésima identificação e Max a função que retorna o maior valor obtido entre o conjunto de atrasos obtidos, então $\mathcal{T}_{Max}^{\mathcal{A}}$ é o atraso máximo do agente \mathcal{A} . Essa relação pode ser obtida por meio da equação 7.7:

$$\mathcal{T}_{max}^{\mathcal{A}} = Max[\mathcal{D}_1(\mathcal{A}), \mathcal{D}_2(\mathcal{A}), \dots, \mathcal{D}_n(\mathcal{A})]$$

Equação 7.8: Atraso máximo experimentado por um agente.

A métrica *Atraso Médio de um Agente* permite identificar o *atraso médio* dos eventos identificados por um agente durante a simulação. Seja $\mathcal{D}_i(\mathcal{A}_j)$ o atraso que um agente \mathcal{A} qualquer da sociedade (o j -ésimo agente) obtem na identificação de um evento e \mathcal{N} o número de eventos identificados, $\mathcal{T}_{Med}^{\mathcal{A}_j}$ é a razão entre a soma dos atrasos ocorridos em cada identificação pelo número de eventos identificados. Nessa equação levamos em consideração apenas os *eventos identificados*, e não os *eventos ocorridos*, pois em algumas situações podem ocorrer dois ou mais eventos ao mesmo tempo. Essa métrica é dada pela equação 7.8:

$$\mathcal{T}_{med}^{\mathcal{A}_j} = \frac{\sum_{n=1}^N \mathcal{D}_j(\mathcal{A})}{\mathcal{N}}$$

Equação 7.9: Atraso médio experimentado por um agente.

A métrica *Atraso Mínimo Entre Todos os Agentes* permite identificar o menor atraso experimentado entre toda a sociedade \mathcal{S} de agentes durante a simulação. Seja $\mathcal{T}_{min}^{\mathcal{A}_j}$ o menor atraso que o agente \mathcal{A}_j ($1 \leq j \leq m$; m é o número de agentes) experimenta durante toda a simulação e *Min* uma função que retorna o menor valor de um conjunto, então o menor atraso obtido entre todos os agentes ($\mathcal{T}_{min}^{\mathcal{S}}$) é dado pela equação 7.9:

$$\mathcal{T}_{min}^{\mathcal{S}} = \text{Min}(\mathcal{T}_{min}^{\mathcal{A}_1}, \mathcal{T}_{min}^{\mathcal{A}_2}, \dots, \mathcal{T}_{min}^{\mathcal{A}_m})$$

Equação 7.10: Atraso mínimo experimentado pela sociedade \mathcal{S} .

A métrica *Atraso Máximo Entre Todos os Agentes* permite identificar o maior atraso experimentado entre toda a sociedade \mathcal{S} de agentes durante a simulação. Seja $\mathcal{T}_{max}^{\mathcal{A}_j}$ o maior atraso que o agente \mathcal{A}_j ($1 \leq j \leq m$; m é o número de agentes) experimenta durante toda a simulação e *Max* uma função que retorna o maior valor de um conjunto, então o atraso máximo entre todos os agentes ($\mathcal{T}_{max}^{\mathcal{S}}$) é dado pela equação 7.10:

$$\mathcal{T}_{max}^{\mathcal{S}} = \text{Max}(\mathcal{T}_{min}^{\mathcal{A}_1}, \mathcal{T}_{min}^{\mathcal{A}_2}, \dots, \mathcal{T}_{min}^{\mathcal{A}_m})$$

Equação 7.11: Atraso máximo experimentado por uma sociedade \mathcal{S} .

A métrica *Atraso Médio de uma Sociedade* permite identificar o atraso médio dos eventos identificados toda a sociedade \mathcal{S} durante a simulação. Seja $\mathcal{T}_{Med}^{\mathcal{A}}$ o atraso médio do agente \mathcal{A} para identificar um conjunto de eventos e $|\mathcal{S}|$ o tamanho da sociedade, $\mathcal{T}_{Med}^{\mathcal{S}}$ é a razão entre a soma dos atrasos médios de cada agente e o tamanho da sociedade, conforme equação 7.11:

$$\mathcal{T}_{med}^{\mathcal{S}} = \frac{\sum_{j=1}^{|\mathcal{S}|} \mathcal{T}_{med}^{\mathcal{A}_j}}{|\mathcal{S}|}$$

Equação 7.12: Atraso médio experimentado por uma sociedade \mathcal{S} .

A métrica *Desvio Padrão do Atraso de uma Sociedade* permite identificar o nível de variação entre o atraso médio de cada agente durante a simulação. Seja $\mathcal{T}_{Med}^{\mathcal{A}}$ o atraso médio do agente \mathcal{A}_j para identificar um conjunto de eventos e $\mathcal{T}_{Med}^{\mathcal{S}}$ o atraso

médio da sociedade, então o desvio padrão dos atrasos médio da sociedade é dado conforme equação 7.12:

$$\mathcal{T}_{stddev}^S = \sqrt{\frac{1}{|S|} \sum_{j=1}^{|S|} (\mathcal{T}_{med}^{\mathcal{A}_j} - \mathcal{T}_{med}^S)^2}$$

Equação 7.13: Desvio padrão dos atrasos médio de uma sociedade \mathcal{S} .

7.5.3 Sucesso Baseado em Atraso

A partir do conceito de atraso, propusemos uma definição particular de sucesso a fim de possibilitar aos pesquisadores uma compreensão da **eficácia** dos agentes.

Especificamente no caso de *eventos evitáveis*, se considerarmos que o evento causal pode ser interrompido a qualquer instante, fazendo com que o *evento principal* não ocorra, então podemos assumir que o valor de sucesso deve ser o mesmo ao longo de todo esse intervalo ($\mathcal{S}(v) = 1$).

Após o termino do evento causal, quando se inicia o evento principal, o valor de $\mathcal{S}(v)$ passa a decrescer linearmente até um instante em que qualquer ação de mitigação será desnecessária ($\mathcal{S}(v) = 0$). Aqui, vale a pena lembrar, nós não estamos interessados apenas em saber se o agente identificou o evento ou não. Mesmo que o agente não consiga chegar a tempo de evitar o evento, sua visita deve ser considerada válida pois se faz necessário que alguma ação seja realizada a fim de mitigar os prejuízos. Em termos práticos, isso quer dizer que mesmo que o evento não tenha sido evitado, existe a necessidade de interrompê-lo, reduzindo os prejuízos. Por exemplo, se um “acidente nuclear” não for evitado, alguma providência deve ser tomada no sentido de minimizar os danos. Sendo assim, não faz sentido ignorar a ação do agente.

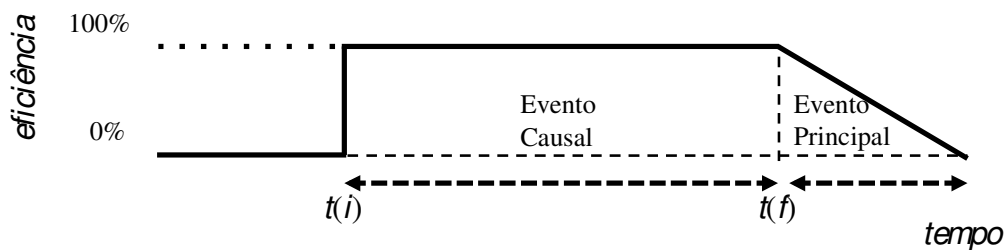


Figura 7.3: Função sucesso para eventos evitáveis.

A partir dessas considerações obtemos então a seguinte função para descrever o sucesso dos agentes em relação aos *eventos evitáveis* conforme ilustrado na equação 7.14.

$$\mathcal{S}_E(\mathcal{V}) = \begin{cases} 1; & \text{se } (t(v) \geq t(i)) \text{ e } (t(v) \leq t(f)) \\ \left| \mathcal{E} * \left[1 - \mathcal{D}(\mathcal{V}) * \left(\frac{L_E}{100} \right) \right] \right|; & \text{se } (t(v) > t(f)) \end{cases}$$

Equação 7.14: Métrica de sucesso para eventos evitáveis.

O valor de $\mathcal{S}_E(\mathcal{V})$ é dado em termos de uma porcentagem de sucesso, proporcional ao momento em que ocorre a visita. Ele é dividido em termos de dois momentos específicos: (i) *durante* o evento causal, e (ii) *depois* do evento causal:

- (i) O agente consegue evitar o evento, ou seja: $(T(f) \leq T(\mathcal{V}) \leq T(i))$. Portanto, o valor de $\mathcal{S}_M(\mathcal{V})$ será igual a 1 (100%);
- (ii) O agente não consegue evitar o evento, mas seu instante de visita ocorre dentro do intervalo de tempo do evento principal ou do evento consequência $(t(v) \leq t_{E_p}(f) \leq t(f))$. Nesse caso, o valor de $\mathcal{S}_M(v)$ será proporcional ao atraso;

Uma vez que a simulação e, consequentemente, a duração dos eventos são definidas em termos de ciclos de simulação (ou turnos), faz-se necessário realizar alguns ajustes. O primeiro diz respeito à *duração de um evento* (L_E) cujo valor é dado em ciclos. Ela é dividida por 100 a fim de ponderar o valor do atraso ($\mathcal{D}(v)$) em relação aos diferentes tamanhos de eventos. Isso significa dizer que um agente que identifica um evento no primeiro ciclo de um evento com 3 ciclos de duração deve receber uma avaliação melhor do que um que identifica um evento com 6 ciclos de duração.

Em seguida, o produto do atraso ($\mathcal{D}(v)$) pelo resultado da divisão de L_E por 100 deve ser subtraído de 1 a fim de tornar o resultado em um valor entre 0 e 1. Uma vez que o requisito por trás dessa métrica é a eficácia não podemos permitir que uma situação como aquela em que $\mathcal{D}(v) = 1$, faça com que $\mathcal{S}_M(v) = 1$, ou seja, o mesmo valor que o agente receberia se tivesse conseguido evitar o evento. Para isso introduzimos a constante de enfraquecimento \mathcal{E} . Seu objetivo é o fazer com que o valor de $\mathcal{S}_M(v)$, mesmo no melhor de caso, seja sempre inferior a 1 ($\mathcal{E} \approx 0.9$).

No caso de **eventos manipuláveis**, o agente deve visitar o vértice durante todo o intervalo de tempo em que o evento principal estiver ocorrendo. Caso não consiga, ele

deve chegar no intervalo de tempo do evento consequência a fim de reduzir os danos. No entanto, nesse caso, quanto mais ele demorar, maiores serão os riscos e eventuais danos colaterais. Por exemplo, no caso do evento “afogamento” o agente pode identificá-lo durante sua ocorrência, evitando que a vítima venha a falecer ($T_A(f)$), mas caso ele ultrapasse esse limite de tempo, a vítima terá sequelas permanentes em decorrência da falta de oxigênio no cérebro mesmo se for resgatada. Ainda assim, o desempenho do agente é considerado, pois, apesar do fracasso em evitar danos à saúde da vítima, ele conseguiu salvar sua vida.

Devido à semelhança entre esse tipo de evento e o anterior, $\mathcal{S}_M(v)$ pode ser calculado utilizando a equação 7.12. Nesse caso $\mathcal{S}_M(v) = 1$ se a visita ocorrer durante a ocorrência do evento principal, e $\mathcal{S}_M(v) < 1$ se ocorrer durante a ocorrência do evento consequência, conforme a figura 7.4.

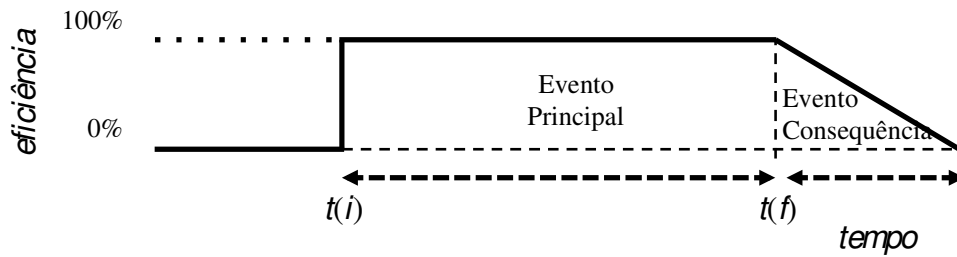


Figura 7.4: Função sucesso para eventos manipuláveis.

No caso de **evento tratável**, devido à duração desprezível ou dificuldade de identificar tanto as causas como o evento principal, a visita só pode ocorrer durante a ocorrência do evento consequência a fim de mitigar os danos. Esse tipo de comportamento pode ser descrito na figura 7.5.

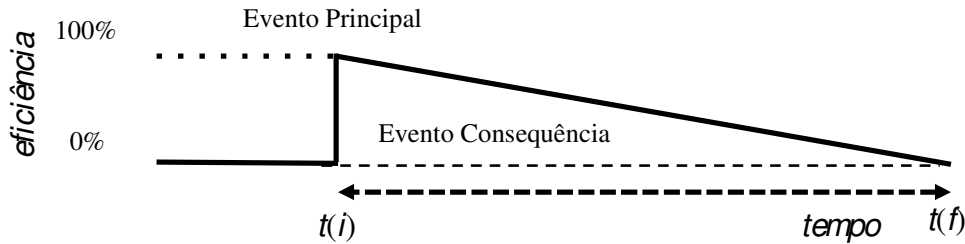


Figura 7.5: Função sucesso para eventos tratáveis.

O sucesso do agente é calculado utilizando a equação 7.15. Nesse caso, se o instante de visita $t(v)$ ocorrer após $t(i)$, a situação (I) ocorre. É importante ressaltar que como nesse tipo de evento não há identificação do evento causal ou principal, então o

agente pode alcançar o máximo de eficácia. Isso ocorre quando o agente visita o vértice no mesmo ciclo de simulação em que o evento se inicia, fazendo com que o valor de $\mathcal{D}(v)$ seja igual a zero. Sendo assim, o uso da constante de enfraquecimento \mathcal{E} se torna desnecessário.

$$\mathcal{S}_T(\mathcal{V}) = 1 - \mathcal{D}(v) * \left(\frac{L_E}{100}\right); \text{ se } (T(\mathcal{V}) > T_A(i))$$

Equação 7.15: Métrica de Sucesso para eventos tratáveis.

7.3 Identificação das Métricas de acordo com Impacto

De acordo com o que foi apresentado na seção 7.1, em alguns tipos de patrulhas, é importante conhecer a capacidade dos agentes em mitigar prejuízos. Conforme apresentado na subseção 5.3.4, todo evento tem um *impacto* o qual representa, de forma simplificada, sua periculosidade.

Essa métrica tem como objetivo avaliar a habilidade dos agentes em identificar eventos de acordo com o impacto de cada um deles. Fator de impacto (\mathcal{F}) representa o quanto de prejuízo um determinado evento pode causar, tanto do ponto de vista *material* como *imaterial*. Sendo assim, \mathcal{F} permite medir a intensidade de cada evento. Uma representação gráfica deste conceito é apresentada na figura 7.6. Ele é constituído de três dimensões: *tempo*, *espaço* e *nocividade*.

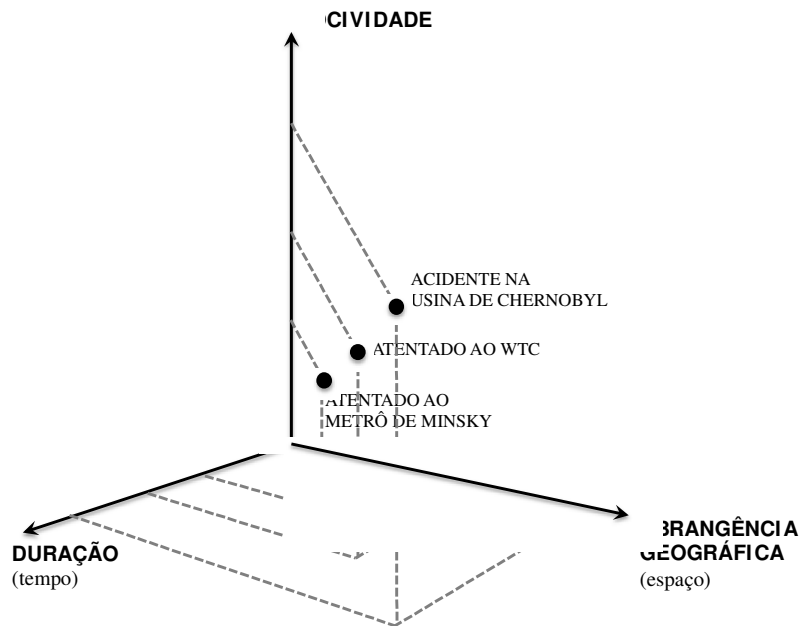


Figura 7.6: Representação do fator de impacto.

No primeiro eixo (*tempo*), temos que, quanto mais duradouro for o evento, maiores tendem a ser suas consequências. No segundo eixo (*espaço*), quanto maior for a área, maior será a quantidade de pessoas e bens materiais afetados. No terceiro eixo (*nocividade*) temos o potencial de dano, que representa a periculosidade de um evento, sobretudo para a vida humana. Por exemplo, um incêndio, por mais abrangente e duradouro que seja, causando a morte de dezenas ou centenas de pessoas, animais e vegetação, não é mais letal do que o vazamento de energia nuclear, pois esse além de ser mais difícil de controlar, pode causar problemas tanto direta como indiretamente. Por outro lado, a disseminação da bactéria causadora do antraz (*bacillus anthracis*), em uma região com as mesmas proporções e pela mesma quantidade de tempo é um evento mais nocivo, pois se não houver o devido tratamento, esta pode levar as vítimas a óbito em alguns horas (Anderson, 2009).

O impacto representa, portanto, o *poder destrutivo* de um determinado evento. Conhecer essa característica é fundamental para os agentes, pois fará com que eles possam dar prioridade àqueles com maior \mathcal{F} .

A fim de ilustrar essa situação, na figura 7.6 podemos observar como três instâncias de eventos reais podem ser representados dentro dessa métrica. No primeiro caso, temos o atentado ao Metrô de Minsky, capital da Bielo-Rússia, em 12 de abril de

2011 onde 11 pessoas morreram devido à explosão de uma bomba no metrô (Guardian, 2011).

No segundo caso, temos o atentado ao *World Trade Center* em Nova Iorque, Estados Unidos, no dia 11 de setembro, onde milhares de pessoas morreram (BBC, 2011). No entanto, apesar da gravidade do evento, sua duração foi bastante limitada e de alcance geográfico de alguns quarteirões nas circunvizinhanças do WTC.

No último caso, o acidente nuclear da usina de Chernobyl ocorrido em 26 de abril de 1986, muitas pessoas morreram imediatamente e milhares ao longo do tempo, por consequências de doenças adquiridas através da poeira radioativa que se espalhou por parte da Europa Oriental e Ocidental. Constatou-se também que uma grande quantidade de crianças nasceu com deformidades devido à exposição direta ou indireta de seus pais à radioatividade. Esse evento além de ser altamente nocivo, teve uma grande duração, ultrapassando várias gerações, e um grande alcance geográfico (INSAG, 1992).

Nós propomos então que para cada evento E , seja realizado um cálculo de \mathcal{F} utilizando a equação 7.16:

$$\mathcal{F}(E) = (D + A + N) + (n.T + m.N)$$

Equação 7.16: Definição de fator de impacto de um evento.

Nela temos que \mathcal{F} é a soma de duas partes: a primeira representa as *propriedades absolutas* do evento, as quais independem do julgamento do pesquisador. Ela é formada pela soma da duração do evento (D), o alcance geográfico (A) e a nocividade (N), explicadas no capítulo 5. Assumimos que todos têm o mesmo peso ou importância. A segunda parte é formada pelas *propriedades relativas* do evento, a qual representa a soma da *Propriedade Tangencial* (T) e da *Propriedade Não-Tangencial* (N). T diz respeito ao valor objetivo das consequências do evento, ou seja, qualquer coisa que pode ser diretamente mensurável financeiramente (sem levar em consideração valores afetivos ou pessoais): um edifício, um automóvel ou uma ponte. Por outro lado, N diz respeito ao valor subjetivo das consequências do evento, aquilo que tem um valor afetivo ou mesmo que tenha um valor difícil de ser quantificado, como uma vida humana, uma obra de arte, um monumento histórico etc. Nesse caso, cada uma terá um peso (n e m) a ser atribuído pelos pesquisadores durante os experimentos.

O sucesso do agente durante toda a patrulha será dado, portanto, pelo somatório dos valores do fator de impacto dos eventos que o agente conseguir identificar em cada visita conforme especificado na equação 7.16.

Nesse caso o pesquisador poderá desenvolver uma série de estratégias interessantes para os agentes. Por exemplo, em uma *estratégia egoísta*, os agentes

procuram identificar uma quantidade menor de eventos, mas que possuam maior fator de impacto. Dessa forma, eles priorizam um vértice em detrimento de outros.

$$\mathcal{S}(\mathcal{V}) = \sum_{e \in E} \mathcal{F}_e$$

Equação 7.16: Sucesso baseado no fator de impacto.

Adicionalmente, \mathcal{F} pode ser utilizada para ponderar os eventos nas métricas apresentadas anteriormente. Nesse caso o pesquisador poderá avaliar ao mesmo tempo como os agentes conseguem equilibrar a eficácia e a mitigabilidade em um mesmo patrulhamento. Na equação 7.17 temos então a função Sucesso para eventos evitáveis ponderados pelo fator de impacto e na equação 7.18 a função Sucesso para eventos tratáveis ponderado pelo fator de impacto.

$$\mathcal{S}_E(\mathcal{V}) = \begin{cases} \mathcal{F}; \text{ se } (t(v) \geq t(i)) \text{ e } (t(v) \leq t(f)) \\ \left| \varepsilon * \left[1 - \mathcal{D}(v) * \left(\frac{L_E}{100} \right) \right] \right| * \mathcal{F}; \text{ se } (t(v) > t(f)) \end{cases}$$

Equação 7.17: Sucesso para eventos evitáveis ponderado pelo fator de impacto.

$$\mathcal{S}_M(\mathcal{V}) = \left[1 - \mathcal{D}(v) * \left(\frac{L_E}{100} \right) \right] * \mathcal{F}; \text{ se } (t(v) > t_A(i))$$

Equação 7.18: Sucesso para eventos tratáveis ponderado pelo fator de impacto.

7.4 Análise Comparativa entre PMAOE e Benchmarks Existentes

Tendo finalizado o conjunto de atividades previstas no PROBENCH, obtivemos os artefatos necessários que compõem PMAOE. Dessa forma, é importante fazermos uma análise comparativa de PMAOE com os benchmarks mencionados na seção 2.4 considerando requisitos elencados na literatura (Drogoul et al., 2007; Huppler, 2009; Stone, 2003) bem como outros advindas da experiência do grupo de pesquisa de SMA do CIn-UFPE.

Problema Atraente: RCS se baseia no futebol, um dos esportes mais populares do mundo. Portanto, ela tem um apelo muito forte, sobretudo entre não-especialistas e empresas que desejam investir como forma de publicidade. RCR lida com um problema muito importante, sobretudo em regiões propensas à desastres naturais, mas não tem despertado o mesmo interesse que RCS, provavelmente por não possuir um apelo de entretenimento. TAC possui um problema interessante e útil, mas se restringe ao contexto de aplicações corporativas. O problema abordado em PMAOE é útil para aplicações de monitoração e vigilância. Em função da importância desses tipos de aplicações e de sua abrangência, PMAOE é importante para o estudo de SMA e robótica;

Fácil Compreensão: conforme dito anteriormente, RCS tem a vantagem de lidar com um problema amplamente conhecido. Portanto, seu funcionamento e regras são bastante conhecidos, o que simplifica sua compreensão e reduz a curva de aprendizado. Por outro lado, faz-se necessário conhecer os detalhes de implementação relacionados à física envolvida no futebol, o que representa um conhecimento adicional a ser adquirido pelo pesquisador. RCR, apesar de também ter um problema bastante conhecido, possui uma grande quantidade de detalhes, restrições e variações, o que dificulta principalmente a replicação de experimentos. No caso de TAC, devido às regras de mercado envolvidas, pode ser mais complexa do que as demais. No caso de PMAOE, o problema consiste em o agente localizar os vértices onde ocorrem ou irão ocorrer eventos a fim de evitá-los ou mitigar os prejuízos. Portanto, nós o consideramos como sendo de fácil compreensão;

Problema Abstrato: os problemas apresentados pelos benchmarks RCS, RCR e TAC são bastante específicos de tal maneira que o desenvolvimento de soluções ficam limitados ao domínio para o qual foram criados. Diferentemente o problema apresentado em PMAOE permite se fazer generalizações das soluções desenvolvidas, o que favorece sua utilização em domínios diferentes. Por exemplo, existem trabalhos de voltados para segurança de redes, resgate de pessoas, vigilância de fronteiras, supervisão de páginas web etc.

Foco na Tarefa: consideramos que tanto RCS como RCR dispersam o foco da tarefa devido a detalhes que não estão relacionados com a exploração de técnicas em SMA, sobretudo nas ligas 3D ou robótica. No caso da liga RCS 2D e RCR existe uma quantidade considerável de detalhes relacionados a questões de projeto tais como visão dos agentes, movimento da bola ou movimento dos jogadores. Eles

fazem com que os pesquisadores invistam esforços em detalhes que não estão relacionados com o objetivo de sua pesquisa. TAC é bastante focado na tarefa em que se propõe (negociação). No caso de PMAOE o problema de patrulhar uma região não demanda esforços por parte do pesquisador em relação a detalhes que não estejam diretamente relacionados com sua solução;

Métricas objetivas: o objetivo em RCS é vencer uma partida de futebol. No entanto, se uma equipe vence, isso não necessariamente significa que a técnica empregada por ela é superior à outra, uma vez que há muitas variáveis envolvidas em uma competição esportiva. Portanto, o resultado do tipo “vencer/perder” é subjetivo, sobretudo quando a diferença de gols é mínima. No caso do RCR as métricas são objetivas (número de vítimas salvas ou número de incêndios apagados), e não precisam de uma interpretação a posterior para compreender o desempenho dos agentes. No caso do TAC as métricas também são objetivas. Conforme visto no capítulo 7, todas as métricas criadas para PMAOE são objetivas;

Métricas claras e simples: Uma partida de RCS possui métricas bastante claras e simples, como por exemplo, quantidade de gols. No caso de RCR, métricas como quantidade de incêndios debelados ou quantidade de pessoas salvas também são claras e simples. TAC possui métricas como quantidade de transações completadas. Alguns exemplos de métricas em PMAOE são quantidade de eventos detectados e atraso médio;

Útil e Realista: RCS demanda esforços significativos na realização de tarefas que não têm utilidade do ponto de vista prático, tais como chutar uma bola corretamente ou impedir que os agentes adversários atinjam seus objetivos (“defender a trave”). Apesar de que a habilidade não tem utilidade que justifique investimentos de pesquisa por parte da comunidade de SMA, mas de toda forma sua realização é essencial para o sucesso da solução (“vencer a partida”). Por razões como essas, vários autores criticam RCS no sentido de que esse benchmark atrapalha a agenda de pesquisas em IA, causando desconfiança por parte do público em geral quanto às possibilidades da área (Lenat, 2008). O problema de RCR é útil e realista e o desenvolvimento de soluções pode atender a demandas importantes, sobretudo para regiões que sofrem com desastres naturais. TAC não é tão importante quanto RCR, mas suas soluções podem contribuir com o desenvolvimento de aplicações de importância econômica. Conforme pudemos observar ao longo desta proposta de tese, as aplicações para

as quais (e consequentemente PMAOE) se destinam são de grande importância, sobretudo aquelas que lidam com segurança, vigilância e resgate de pessoas;

Evolucionável: Conforme vimos no capítulo 4.4, uma forma de constatar que é evolucionável é observando que inicialmente havia apenas a MATP, depois surgiram trabalhos voltados para AMAP. RCS também tem evoluído ao longo dos anos, mas apenas no sentido de aperfeiçoar seu funcionamento (mantendo o problema sem alterações). RCR tem evoluído no sentido de disponibilizar desafios mais sofisticados e realistas. Por fim, TAC tem evoluído significativamente, sendo disponibilizadas atualmente vários formatos da competição conforme vimos na seção 2.4.3 TAC.

Quadro 7.1: Análise comparativa dos benchmarks em SMA.

	RCS	RCR	TAC	PMAOE
Problema atraente	++	++	+	++
Fácil compreensão	+	+	+	++
Problema abstrato	+	+	+	++
Foco na tarefa	-	-	++	++
Métricas objetivas	-	+	++	++
Métricas claras e simples	++	++	++	++
Útil e realista	-	++	++	++
Evolucionável	-	-	++	++

Legenda: (++): atende fortemente, (+): atende minimamente, (-): não atende.

7.6 Conclusões sobre o Capítulo

Neste capítulo apresentamos as métricas que serão utilizadas em PMAOE como forma de avaliar o desempenho dos agentes. Essas métricas foram definidas seguindo as atividades da etapa de definição de métricas de PROBENCH, conforme especificado no capítulo 3.

As métricas foram organizadas através de um conjunto de requisitos, onde cada uma descreve um objetivo diferente na forma como experimentos e, consequentemente, avaliações comparativas devem ser conduzidas.

O fato de que na PMAOE lidamos com uma nova proposta para o problema, que é a de tornar agentes munidos de conhecimento do mundo, levando em consideração, sobretudo a natureza dos eventos, nos impôs a necessidade de criarmos métricas totalmente novas. As métricas foram desenvolvidas visando os seguintes critérios-objetivos:

- Maximizar a quantidade de eventos identificados;
- Minimizar o atraso na identificação de eventos;

- Minimizar a quantidade de prejuízo decorrente dos eventos ocorridos;

Apesar de que apresentamos um conjunto abrangente de métricas, contemplando diferentes aspectos dos eventos, acreditamos que outras ainda podem ser criadas a fim de estender o presente trabalho. Nesse sentido, é importante ressaltar, que mais importante do que a criação das métricas foi a definição requisitos os quais podem auxiliar pesquisadores a criar outras novas.

Sendo assim, consideramos que neste capítulo realizamos duas importantes contribuições: na primeira pudemos demonstrar a viabilidade de PROBENCH através das diretrizes da etapa denominada “Definição das Métricas” e na segunda identificamos um conjunto de requisitos para o problema da patrulha, através dos quais criamos novas métricas para a realização de estudos comparativos em SMA.

No próximo capítulo iremos apresentar uma visão geral da tese, ressaltando quais objetivos já foram alcançados e quais atividades precisam ser realizadas a fim de finalizá-la.

Capítulo 8

Estudo Experimental

Nos capítulos anteriores foi demonstrado de que forma as etapas do PROBENCH foram conduzidas a fim de construirmos o benchmark PMAOE. Este capítulo apresenta um conjunto de experimentos a fim de demonstrar sua viabilidade prática. Cada experimento explora um *dataset* diferente e faz uso de agentes previamente estudados na literatura abordando o problema da patrulha. Dessa forma, será possível analisar como estratégias projetadas para lidar com ociosidade se comportam nesse novo contexto e de que forma conceitos introduzidos nesse trabalho podem influenciar no comportamento dos agentes.

8.1 SIMPATROL

A ferramenta utilizada na realização dos experimentos deste trabalho foi SIMPATROL (Moreira, 2008; Araújo, 2010), um simulador de SMA voltado exclusivamente para o problema da patrulha e já utilizado em diversos trabalhos realizados pelo grupo de pesquisa em SMA do CIn-UFPE (Moreira, 2008). As justificativas para sua escolha foram as seguintes:

- **reuso de experiência:** uma vez que a ferramenta foi desenvolvida no contexto dessa tese, seria mais fácil obter recursos e informações a respeito de seu funcionamento;
- **disponibilidade dos elementos básicos:** uma vez que já dispunhamos de uma ferramenta que dispunha dos elementos necessários para o estudo de problemas básicos da patrulha, resolvemos estendê-la com as novas características.

Conforme vimos na seção

5.3 Modelagem da Classe de Problema, PMAOE introduz vários conceitos e comportamentos novos, o que impõe modificações ao SIMPATROL. Essas modificações estão relacionadas com o mecanismo de deliberação dos agentes (seção 5.2.4 Exemplo Ilustrativo), geração de eventos (seção 5.3.4 Exemplos de Eventos) e avaliação de desempenho por meio de métricas adequadas (capítulo 7).

8.2 Agentes Estudados

Nos experimentos realizados foram utilizados três tipos de agentes previamente estudados pelo grupo de pesquisa em SMA do CIn-UFPE: *Cognitive Coordinator* (CC) (REF), *Heuristic Pathfinder Cognitive Coordinated* (REF) e *Gravitational Coordinator* (Sampaio et al., 2011).

O agente *cognitivo coordenado* (CC) consiste em uma arquitetura de agentes cognitivos, possuindo um coordenador central que é responsável por distribuir os vértices entre os agentes. Para isto é essencial a existência de um sistema de comunicação baseado em troca de mensagens (Machado, 2002). O coordenador central, que possui acesso às ociosidades compartilhadas do grafo, realiza esta distribuição dos vértices oferecendo a cada patrulheiro o vértice de maior ociosidade no momento que ainda não é objetivo de nenhum agente. Cada agente deve então, ao receber seu vértice objetivo, planejar uma rota até ele utilizando algum algoritmo de distâncias mínimas, como Dijkstra, onde os pesos das arestas representam as distâncias entre as regiões de interesse.

O agente *cognitivo coordenado localizador de rotas heurístico* (HPCC) é baseado no CC, mas difere no fato de que a escolha de suas rotas leva em consideração a relação entre distância e ociosidade dos vértices da rota. Isso significa que o agente coordenador escolherá preferencialmente para cada agente coordenado rotas mais longas desde que esta propicie a visita a vértices com maior ociosidade.

Finalmente, o agente *cognitivo coordenado localizador de rotas heurístico orientado a eventos* (EOHPCC) foi desenvolvido no contexto deste trabalho a partir do HPCC. O seu mecanismo de decisão das rotas a serem percorridas visa contemplar o conceito de evento aqui introduzido. Dessa forma, quando não houver eventos ocorrendo nos vértices que compõem a rota sendo analisada, o coordenador se comporta como o HPCC. Caso contrário, ele considera o fator de impacto do evento ocorrendo.

8.3 Metodologia

O estudo experimental foi realizado utilizando 3 tipos de agentes conforme explicado na seção 8.2 em três tipos de cenários diferentes. Cada um desses foi representando por um *dataset* diferente e sua escolha foi feita de forma a explorar diferentes níveis de complexidade, indo do caso mais simples até o mais complexo. Em cada experimento, foram realizadas 10 simulações para cada tipo de agente, cada uma tendo uma duração de 5000 ciclos. Ao final dos experimentos foi calculada a média dos resultados obtidos para cada uma das métricas.

8.4 Experimento 1: Afogamento

8.3.1 Dataset

No primeiro experimentos, os agentes devem ser capazes de identificar a ocorrência de eventos do tipo *afogamento* restritos a determinados vértices que representam piscinas, praias, rios etc. Conforme pudemos ver na seção 5.3.4 Exemplos de Eventos não é possível evitar que um evento dessa natureza seja iniciado, devendo o agente, portanto, detectá-lo no momento de ocorrência do evento principal. Da mesma forma, se o agente chegar ao vértice após o término do evento, a vítima terá ido a óbito.

Outra característica diz respeito ao grafo escolhido. O mapa utilizado é o “circular” no qual existem apenas duas opções de visita em cada vértice. Esse é um grafo cuja deliberação a respeito de qual vértice (e, conseqüentemente, qual rota) deve ser escolhida é bastante simples.

A seguir iremos apresentar os valores escolhidos e a motivação associada a cada um para o *grafo*, *sociedade*, *agente* e *evento*:

- **Grafo:** conforme visto na seção 5.3.3 Taxonomia da Classe de Problematemos três atributos associados ao *vértice*: *disponibilidade*, *prioridade* e *duração da visita*. Uma vez que esse é um trabalho inicial, optamos por usar grafos sem dinamicidade a fim de termos maior controle sobre os experimentos. Uma vez que em PMAOE a *prioridade* está associada aos eventos, e não aos vértices, não adotamos nenhum valor especial para este atributo. O valor da *duração da visita* foi assumido como *default*, pois esse atributo também não é importante para os objetivos do experimento. Quanto ao *arco*, o custo é heterogêneo, pois a ligação entre os vértices depende de fatores como distância ou preço. Uma vez que acreditamos que a *direção* tem implicações sobre o algoritmo dos agentes, optamos pela unidirecionalidade, deixando para

trabalhos posteriores analisarmos essa opção. Da mesma forma que no caso do vértice, os arcos são estáticos e, portanto, eles estão sempre disponíveis. Por fim, o *tamanho do grafo* varia de um tipo para outro, mas sempre se mantém o mesmo em relação entre as diferentes simulações;

- **Sociedade:** *organização* é heterogênea devido ao fato de que nas estratégias utilizadas há sempre dois tipos de agentes: um coordenador e um ou mais coordenados. Pelo mesmo motivo do grafo, adotamos uma sociedade sem dinamicidade. A *comunicação*: é realizada através da troca de mensagens entre o agente coordenador e os agentes coordenados. No entanto, os agentes coordenados não se comunicam entre si. Por fim, foram escolhidos dois tamanhos de sociedades: 6 e 11;
- **Agente:** uma vez que o agente coordenado recebe as informações do coordenador, então, tanto em relação à sociedade como ao grafo, o coordenador possui *informação a priori* enquanto o coordenado não possui. Em relação ao *alcance da percepção*, o agente coordenador tem percepção completa do grafo. Quanto à confiabilidade da percepção, não estamos considerando a existência de ruído em função da imprecisão dos sensores ou de quaisquer outros fatores.
- **Métricas:** as métricas a serem utilizadas serão aquelas apresentadas nas seções 7.3 e 7.4: porcentagem de eventos detectados, porcentagem de eventos evitados, porcentagem de eventos manipulados, porcentagem de eventos recuperados, atraso mínimo, médio e máximo de detecção de eventos.

O Quadro 8.1: Dataset utilizado no experimento 1. apresenta de forma resumida as informações referentes ao *dataset*.

Quadro 8.1: Dataset utilizado no experimento 1.

Grafo	Tipo	Círcular, Corredor, Ilha	
	Vértice	Disponibilidade	Sim
		Prioridade	Não
		Duração da visita	Irrelevante
	Arco	Custo	Heterogêneo
		Direção	Unidirecional
		Disponibilidade	Sim

	Tamanho	Heterogêneo		
Sociedade	Organização	Heterogênea		
	Dinamicidade	Estática		
	Comunicação	Troca de mensagens		
	Tamanho	6, 11		
Agente	Informação a Priori	Sociedade	Coordenador: sim Coordenado: não	
		Grafo	Coordenador: sim Coordenado: não	
	Percepção	Objeto	Grafo	Coordenador: sim Coordenado: não
			Sociedade	Coordenador: sim Coordenado: não
			Custo do arco	Coordenador: sim Coordenado: não
			Evento	Coordenador: sim Coordenado: não
			Tempo	Coordenador: sim Coordenado: não
		Alcance	Coordenador: Grafo	
		Confiabilidade	Sim	
Evento	Tipo		DROWNING	
	Categoria		MAIN	
	Origem		ACCIDENT_HUMAN	
	Ocorrência	Turno	Específico	
		Nodo	v10,v25,v30,v40	
	Pré-condição		-	
	Impacto	Prejuízo	Tangível	1
			Intangível	10
		Abrangência		1
		Nocividade		5

	Duração	6
	Atendimento	EVT_MANAGEABLE
	Consequências	-
Métrica	Eficiência	Eventos detectados, eventos manipulados, eventos recuperados
	Responsividade	Atraso médio

8.3.2 Resultados e Discussão

O quadro 8.2 mostra os resultados do experimento para cada um dos agentes utilizados. Em cada coluna apresentamos o valor da porcentagem média dos eventos detectados para cada tamanho de sociedade utilizada. A partir dela é possível constatar que, apesar dos três tipos de agentes possuírem características bastante comuns, há uma grande diferença nos resultados obtidos. Inicialmente podemos perceber que CC não detectou nenhum evento tanto com 6 como com 11 agentes. Em relação às demais estratégias, pudemos observar que o aumento da sociedade teve um efeito considerável no desempenho (um aumento de 54,83% no caso de HPCC e 30,46% no de EOHPCC). Apesar de que EOHPCC dispõe de informação sobre o *fator de impacto* dos eventos, quando a população utilizada foi de 11 agentes, HPCC teve um melhor desempenho.

Quadro 8.2: porcentagem média dos eventos detectados.

	6 agentes	11 agentes
CC	0,0	0,0
HPCC	30,71	68,00
EOHPCC	44,05	63,35

Em relação ao atraso médio dos eventos detectados (Quadro 8.3: atraso médio dos eventos detectados.), podemos observar um comportamento semelhante: quando o tamanho da população utilizado foi de 6 agentes o desempenho de EOHPCC foi superior (*atraso médio* = 1,04), enquanto que quando o tamanho foi de 11 agentes, HPCC teve melhor desempenho (*atraso médio* = 0,87).

Quadro 8.3: atraso médio dos eventos detectados.

	6 agentes	11 agentes
CC	--	--
HPCC	2,93	0,49
EOHPCC	1,04	0,87

8.4 Experimentos 2: Colisão de Automóveis

8.4.1 Dataset

No segundo experimento, os agentes devem detectar a ocorrência de eventos do tipo *colisão de automóveis*. Similar ao caso anterior, esse evento não é evitável. No entanto, uma vez que a duração do evento principal é desprezível, resta à cada agente detectar o evento consequência. Esse evento pode ocorrer em qualquer lugar, devendo o agente dar a mesma atenção a todos os vértices do grafo. Por essa razão, a fim de evitar uma grande quantidade de eventos ocorrendo ao mesmo tempo, reduzimos a probabilidade de ocorrência.

O grafo utilizado nesse experimento foi um *mapa A* com 50 vértices conforme apresentado na seção 4.3.2 Região

O quadro 8.4 apresenta os atributos do *dataset* com os valores utilizados no primeiro experimento:

Quadro 8.4: Dataset utilizado no experimento 1.

Grafo	Tipo	Mapa A	
	Vértice	Disponibilidade	Sim
		Prioridade	Não
		Duração da visita	-
	Arco	Custo	Heterogêneo
		Direção	Unidirecional
		Disponibilidade	Sim
	Tamanho	Heterogêneo	
Sociedade	Organização	Heterogênea	
	Dinamicidade	Estática	
	Comunicação	Troca de mensagens	
	Tamanho	6, 11	
Agente	Informação a	Sociedade	Coordenador: sim

	Priori			Coordenado: não	
		Grafo		Coordenador: sim Coordenado: não	
	Percepção	Objeto	Grafo	Coordenador: sim Coordenado: não	
			Sociedade	Coordenador: sim Coordenado: não	
			Custo do arco	Coordenador: sim Coordenado: não	
			Evento	Coordenador: sim Coordenado: não	
			Tempo	Coordenador: sim Coordenado: não	
		Alcance		Coordenador: Grafo	
Confiabilidade		Sim			
Evento	Tipo			CAR_CRASH	
	Categoria			MAIN	
	Origem			MACHINE_ACCIDENT	
	Ocorrência	Turno		Qualquer um.	
		Nodo		--	
	Pré-condição			-	
	Impacto	Prejuízo	Tangível	2	
			Intangível	8	
		Abrangência		0	
		Nocividade		1	
	Duração			1	
Atendimento			EVT_RECOVERABLE		
Consequências			-		
Métrica	Eficiência			Eventos detectados	
	Responsividade			Atraso médio	

8.4.2 Resultados e Discussão

O quadro 6.2 mostra os resultados do experimento 2. Em termos gerais podemos observar um comportamento semelhante ao do experimento 1 com a diferença que desta vez CC detectou uma quantidade razoável de evento quando comparado com HPCC e EOHPCC. No entanto, quando o tamanho foi de 11 agentes seu desempenho foi bem abaixo comparado aos demais. Adicionalmente, HPCC e EOHPCC tiveram um desempenho consideravelmente inferior ao observado no experimento 1. Isso era previsto uma vez que o *dataset* desse experimento foi escolhido com o objetivo de ser mais difícil de patrulhar.

Em relação ao desempenho geral no experimento 2, quando o tamanho da patrulha foi de 6 agentes, HPCC obteve um melhor resultado enquanto que quando tem 11 agentes, o inverso ocorre.

Quadro 8.5: porcentagem média dos eventos detectados.

	6 agentes	11 agentes
CC	0.96	1.88
HPCC	11.58	12.39
EOHPCC	8.86	13.75

No quadro 8.6 podemos observar que quando o tamanho da patrulha é de 6 agentes, EOHPCC obteve o menor atraso médio (1,04), e quando é de 11 agentes é HPCC que obteve o menor atraso (0,49). É importante observar também que, em relação à porcentagem de eventos detectados, não existe relação entre essa métrica e o atraso médio. Portanto, o fato de uma estratégia ser mais eficiente, não garante que ele tem uma maior responsividade.

Quadro 8.6: atraso médio dos eventos detectados.

	6 agentes	11 agentes
CC	5,23	0,75
HPCC	2,93	0,49
EOHPCC	1,04	0,87

8.5 Experimento 3: Explosão em Caldeira

8.5.1 Dataset

No último experimento, os agentes devem detectar a ocorrência de eventos do tipo *explosão de caldeira*. Esse tipo de evento possui um evento causal (*temperatura aumentando*) passível de ser detectado, permitindo ao agente evitar a ocorrência do evento principal (*explosão da caldeira*). Este evento principal, por sua vez, tem duração desprezível, mas possui um evento consequência (*incêndio*) o qual deve ser detectado o mais cedo possível. O grafo utilizado foi “ilha”. Os demais atributos possuem os mesmos valores dos cenários anteriores.

Quadro 8.7: Dataset para o experimento 3.

Grafo	Tipo	Ilha		
	Vértice	Disponibilidade		Sim
		Prioridade		Não
		Duração da visita		-
	Arco	Custo		Heterogêneo
		Direção		Unidirecional
		Disponibilidade		Sim
	Tamanho	Heterogêneo		
Sociedade	Organização	Heterogênea		
	Dinamicidade	Estática		
	Comunicação	Troca de mensagens		
	Tamanho	6, 11		
Agente	Informação a Priori	Sociedade	Coordenador: sim Coordenado: não	
		Grafo	Coordenador: sim Coordenado: não	
	Percepção	Objeto	Grafo	Coordenador: sim Coordenado: não
			Sociedade	Coordenador: sim Coordenado: não
			Custo do	Coordenador:

			arco	sim Coordenado: não
			Evento	Coordenador: sim Coordenado: não
			Tempo	Coordenador: sim Coordenado: não
		Alcance	Coordenador: Grafo	
		Confiabilidade	Sim	
Evento	Tipo		CAR_COLLISION	
	Categoria		MAIN	
	Origem		MACHINE_ACCIDENT	
	Ocorrência	Turno	Qualquer	
		Nodo	Qualquer	
	Pré-condição		-	
	Impacto	Prejuízo	Tangível	2
			Intangível	8
		Abrangência		3
		Nocividade		1
	Duração		5	
	Atendimento		EVT_RECOVERABLE	
	Consequências		FIRE_EVT	

8.5.2 Resultados e Discussão

Nesse experimento não foi possível identificar nenhum evento, e consequentemente não temos nenhum atraso médio. Assim como no experimento anterior, esse resultado era esperado visto que esse era o *dataset* com maior dificuldade de patrulhamento, sobretudo no que diz respeito ao mapa utilizado.

Apesar desse resultado, consideramos que esse cenário é interessante do ponto de vista da patrulha, pois representa um conjunto de situações importante do mundo real. Podemos citar como exemplo o patrulhamento de redes de computadores, departamentos ou cidades, cujas atividades de interesse são realizadas em locais cujas distâncias e/ou custo de interligação podem comprometer o desempenho do sistema como um todo.

Conclusões sobre o Capítulo

Neste capítulo foi apresentado um estudo experimental do problema da patrulha por meio do benchmark PMAOE. Para isso foram realizados três experimentos diferentes. No primeiro, foi analisado o comportamento dos agentes CC, HPCC e EOHPCC quando patrulhando um mapa circular em busca de eventos do tipo afogamento. No segundo, os agentes foram analisados durante a patrulha de um mapa A em busca de eventos do tipo colisão de automóveis. Por fim, no terceiro experimento, os agentes patrulharam uma mapa do tipo ilha em busca de explosões de caldeiras.

Através dos experimentos realizados, foi possível observar que PMAOE pode contribuir para a realização de estudos em SMA. Devido ao fato de que os agentes são submetidos a situações nas quais é possível dispor de informações sobre o ambiente, pudemos analisar o desempenho de “estratégias de busca cega” sob uma perspectiva baseada em conhecimento.

Nesse sentido, podemos concluir a partir dos resultados obtidos em cada experimento que as estratégias baseadas em ociosidade possuem desempenho limitado na tarefa de detectar eventos. Portanto, faz-se necessário investigar e desenvolver novas estratégias para o problema da patrulha sob essa nova perspectiva.

Capítulo 9

Conclusões e Trabalhos Futuros

Este trabalho explorou a idéia de que benchmark representa uma abordagem adequada para a realização de experimentos em SMA. Ele é motivado principalmente pela constatação de que a pesquisa científica nesse campo vem sendo realizada de forma isolada e sem compartilhamento dos dados e do aparato experimental, o que compromete a validade dos resultados obtidos.

Ao longo dos capítulos fomos capazes de demonstrar de forma sistemática como obter os artefatos necessários à realização de benchmarks em SMA. Em função dessa necessidade, desenvolvemos um processo *ad hoc* que auxilie o desenvolvimento de benchmarks. A razão disto é que benchmarks são métodos complexos demais para serem desenvolvidos informalmente. Além disso, a demanda crescente por esse tipo de abordagem impõe a necessidade de meios eficientes e rigorosos que possibilite aumentar a produtividade dos pesquisadores e a qualidade dos benchmarks durante sua criação.

O objetivo deste capítulo é fazer uma descrição resumida das atividades que foram apresentadas ao longo desse trabalho a fim de ressaltar os desafios propostos, identificar o estágio da pesquisa em que nos encontramos e, por fim, apresentar quais atividades podem ser realizadas como forma de dar continuidade a esse trabalho.

9.1 Resumo do Trabalho

Conforme discutido no capítulo 2, benchmarks padronizados permitem avaliar o avanço do estado da arte de uma determinada área de pesquisa. No contexto dos sistemas computacionais, benchmarks tem demonstrado ser um método de grande importância para a condução de experimentos. A exemplo do que ocorre em outras áreas, sua utilização desempenha grande importância para promover a qualidade dos processos e produtos através da comparação dos resultados obtidos por soluções concorrentes.

Com o objetivo de contribuir com a área de SMA, investigamos de que forma benchmarks tem sido utilizados na condução de suas pesquisas. Nossa conclusão é que

há diversos grupos trabalhando isoladamente, sem compartilhamento de cenários, resultados, métricas e ferramentas. Adicionalmente, observamos que não existem processos, técnicas ou diretrizes que auxiliem no desenvolvimento de novos benchmarks.

Devido à sua importância no contexto da metodologia da pesquisa científica (Sim, 2003), discutimos como benchmarks podem contribuir como instrumento de validação científica em IA, mais especificamente em SMA. Nesse sentido, analisamos também como as diferentes combinações de cenários e soluções influenciam na compreensão do sistema em estudo.

Como forma de demonstrar sua viabilidade, desenvolvemos um benchmark abordando o problema da patrulha, ou simplesmente Patrulha Multiagente. Esse problema tem recebido atenção crescente nos últimos anos tanto da comunidade de SMA como de robótica por permitir explorar habilidades tais como negociação, comunicação, autonomia, cooperação etc., bem como atender um grande número de aplicações de mundo real envolvendo a tarefa de vigilância e monitoração.

A fim de suprir essa lacuna, propomos e descrevemos um processo de criação de benchmarks denominado de PROBENCH, o qual é organizado através de um conjunto de fases, cada uma com o objetivo de obter os artefatos necessários para a realização de experimentos envolvendo SMA.

Apesar de seu potencial como classe de problema em benchmarks, apresenta algumas limitações: (i) *falta de um entendimento claro e consensual do que é como classe de problema*, o que faz com que a pesquisa atual seja confusa e desorganizada; (ii) *impossibilidade de comparação*, a qual se deve ao fato de que cada trabalho segue abordagens distintas e não há critérios de comparação, e (iii) a *super-simplificação dos problemas estudados*, diz respeito à adoção de cenários que não correspondem à realidade dos problemas aos quais se pretende aplicar o patrulhamento baseado em agentes.

A fim de superar essas limitações e possibilitar a adoção de como classe de problema em um benchmark, realizamos uma organização conceitual. Em primeiro lugar, criamos uma taxonomia do problema da patrulha durante a fase de *Modelagem da Classe de Problema*, detalhando seus elementos básicos e como eles se relacionam. Essa taxonomia tem as seguintes vantagens: simplifica o entendimento do problema da patrulha; auxilia na criação de cenários; e permite identificar oportunidades de pesquisa. Em segundo lugar, ao estender o problema da patrulha por meio da introdução do conceito de evento, fomos capazes de unificar as diferentes abordagens sob uma mesma perspectiva, simplificando sua compreensão e aumentando as possibilidades de pesquisas. Sendo assim, podemos afirmar que toda patrulha é orientada a eventos: no caso da MATP, o agente não tem conhecimento explícito de natureza dos eventos, e no

caso da MAAP, o evento a ser identificado pelo agente consiste na invasão de lugares específicos do grafo por parte de agentes adversários.

Nesse novo tipo de patrulha, denominada de Patrulha Multiagente Orientada a Eventos (OEMAP), os agentes sabem onde pode ocorrer cada tipo de evento e quais circunstâncias favorecem essa ocorrência. Dessa maneira, eles podem planejar suas rotas levando em consideração essa informação e assim aperfeiçoar seu comportamento.

Uma vez que eventos constituem um conceito central em PMAOE, realizamos um estudo aprofundado acerca do termo no contexto da Metafísica, a qual é a área da Filosofia que se preocupa com as “primeiras coisas” ou “conceitos imateriais”, dentre eles os eventos (Aristoteles, 2010; Davidson, 1980; Kim, 1993; Brand, 1997; Garret, 2006; Carati e Varzi, 2010; Scheneider, 2010). Esse estudo nos permitiu compreender o que é um evento independente do domínio de conhecimento, e como ele pode ser adotado no contexto do problema da patrulha. Com o objetivo de representar de forma estruturada o conhecimento acerca desse conceito e possibilitar seu uso na criação de cenários, criamos uma taxonomia de eventos.

A partir daí instanciamos as taxonomias apresentadas e criamos os cenários para a realização de experimentos em OEMAP de acordo com requisitos tais como simplicidade e dinamicidade. Em seguida definimos um conjunto de novas métricas que permita avaliar a habilidade dos agentes em detectar eventos de acordo com requisitos tais como *eficiência*, *responsividade* e *mitigabilidade*.

Em relação à ferramenta de simulação, optamos por adotar o SIMPATROL, um simulador de SMA específico para o problema da patrulha desenvolvido no CIn-UFPE. Uma vez que esse simulador não lida com eventos, nossa próxima etapa será modificá-lo a fim de que ele possa incluir tanto a geração de eventos como os cenários e as métricas propostas. A descrição da atividade de escolha dessa ferramenta será apresentada na versão final dessa tese, juntamente aos resultados dos experimentos realizados.

9.3 Principais Contribuições

Através do presente trabalho fomos capazes de alcançar três importantes contribuições para o estudo de SMA, sobretudo em relação à PMA.

A primeira foi a organização conceitual do problema da patrulha. Apesar de haver uma grande quantidade de trabalhos publicados e um interesse crescente sobre o tema por parte da comunidade de SMA e robótica, percebemos haver uma desorganização considerável. Isso pôde ser observado quando analisamos a variedade de descrições diferentes do problema na literatura bem como a ausência de uma definição adequada. Uma parte importante dessa organização foi obtida por meio da taxonomia de . Além de auxiliar na caracterização da classe de problema, ela permite

distinguir e agrupar diferentes instâncias de acordo com o objetivo de estudo. Também, ela sintetiza a descrição do problema por meio de um vocabulário padronizado, favorecendo a replicação dos experimentos e a comparação dos resultados obtidos.

A segunda contribuição desta pesquisa é o desenvolvimento do benchmark para SMA denominado PMAOE. Inicialmente observamos que existem poucas abordagens com o objetivo de avaliar soluções baseadas em SMA de forma padronizada ou compartilhada. Estas consistem em competições, o que não é adequado como método de avaliação do progresso científico de uma área do conhecimento. Nessa tese chamamos a atenção da comunidade de SMA para esta importante problemática e apontamos caminhos que podem ser utilizados na sua solução. O benchmark foi desenvolvido a partir de uma classe de problema denominada , a qual vem recebendo crescente atenção da comunidade de SMA e de Robótica. Ela abrange domínios como aplicações militares, supervisão de patrimônio, localização de pessoas perdidas e gerência de redes de computadores. também permite o estudo de comportamentos inteligentes importantes, tais como planejamento, negociação, coordenação, comunicação, cooperação, aprendizado e tomada de decisão. Ao estendê-lo para lidar com eventos, criando a PMAOE, contribuímos para o desenvolvimento de soluções de patrulhamento coerentes com as demandas do mundo real e unificamos os diferentes tipos de patrulhas.

A terceira contribuição é a criação de um processo para o desenvolvimento de benchmarks para SMA. Apesar de sua importância como método de experimentação nessa importante área da Ciência da Computação, não conseguimos localizar na literatura nenhum trabalho abordando metodologias, técnicas, processos ou diretrizes que atendam à essa demanda.

9.3 Trabalhos Futuros

As atividades futuras deste trabalho consistem em ampliar e tornar disponível o PMAOE para a comunidade de pesquisadores em SMA. Dessa forma, pretendo realizar o seguinte:

- *Estender os eventos da PMAOE a fim de inserir causalidade e consequência:* nesse caso será possível realizar estudos mais amplos acerca da capacidade de detecção de eventos. Será possível explorar, por exemplo, a capacidade dos agentes evitarem a ocorrência dos eventos;
- *Implementar um modelo mais realista de ocorrência dos eventos:* na implementação atual do SIMPATROL, os eventos ocorrerem dentro de intervalos pré-definidos com uma determinada probabilidade. A caracterização dos eventos como sendo processos estocásticos pode levar a um modelo mais coerente com o mundo real;

- *Testar os agentes em cenários mais complexos*: de acordo com o estudado nos capítulos 4 e 6, os agentes podem ser estudados mediante várias perspectivas tais como *sociedade*, *grafo* e *eventos*, onde cada uma apresenta diversos atributos. É importante, portanto, fazer experimentos no sentido de explorar o comportamento de cada estratégia mediante variações em cada um desses atributos;
- *Disponibilizar o acesso ao SimPatrol via web*: o objetivo dessa linha de trabalho é permitir que pesquisadores em SMA desenvolvam seus agentes e submetam a um conjunto de testes padrão e disponibilizar os resultados de tal forma que outros pesquisadores conheçam os resultados alcançados pelos seus pares. Acreditamos que essa *gamificação* do domínio de problema irá despertar a atenção de novos pesquisadores para o problema e estimulará a comunidade existente a aperfeiçoar suas soluções;
- *Estender outras estratégias a fim de acomodar a detecção de eventos*: estratégias existentes na literatura tais como *Reativo Cognitivo* (Machado et al., 2002), *Cognitivo Coordenado* (Almeida, 2003) e *Ciclo Único* (Chevaileyre, 2004) tem sido estudados em vários trabalhos. É interessante analisar como suas estratégias se comportam nesse novo tipo de perspectiva;
- *Implementar uma estratégia especificamente voltada a eventos*: uma vez conhecidos os comportamentos das estratégias existentes no contexto de cenários baseados em eventos, faz-se necessário explorá-los por meio de
- *Desenvolver outros benchmarks usando PROBENCH*: a fim de comprovar a generalidade do processo *ad hoc* desenvolvido nesse trabalho, é fundamental explorá-lo na criação de benchmarks para outras subáreas da IA, tais como Aprendizagem, Mineração de Textos, Agentes Afetivos etc.

Referências

- Aristóteles (2010). *Categorias: Tradução do Grego Clássico*. Tradutor: José Veríssimo Teixeira da Mata. 2ª Edição. São Paulo: Mantin-Claret.
- Abate, F. R. (1996). *The Oxford Dictionary and Thesaurus: The Ultimate Language Reference for American Readers*. Oxford University Press.
- ABNT (2008). *NBR ISO 9001: Sistemas de Gestão da Qualidade - Requisitos*. Associação Brasileira de Normas Técnicas. Segunda Edição, Rio de Janeiro, 2008.
- Agmon, N., Kaminka, G. A., & Kraus, S. (2008). Multi-robot fence patrol in adversarial domains. In *Proceedings of the Tenth Conference on Intelligent Autonomous Systems (IAS-10)*, pp. 193-201. IOS Press.
- Agmon, N., Kraus, S., & Kaminka, G. A. (2008). Multi-robot perimeter patrol in adversarial settings. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Agmon, N., Kraus, S., Gaminka, G. (2009). Uncertainties in Adversarial Patrol. *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-09)* – vol. 2, pp. 1267-1268.
- Agmon, N. (2010). On Events in Multi-Robot Patrol in Adversarial Environments. In: *Proc. of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds). May, 10 – 14, 2010, Toronto, Canada, pp. 591 – 598.
- Ahmadi, M. e Stone, P. (2005). Continuous Area Sweeping: A Task Definition and Initial Approach. In: *The 12th International Conference of Advanced Robotics*, July, 2005.
- Ahmadi, M., & Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Albayraktaroglu, K., Jaleel, A., Wu, X., Franklin, M., Jacob, B., Tseng, C., Yeung, D. (2005). BioBench: A Benchmark Suite of Bioinformatics Applications. In: *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, 2005. ISPASS 2005.
- Bogdan Alexe, Wang-Chiew Tan, Yannis Velegrakis. STBenchmark: Towards a Benchmark for Mapping Systems. In *Proceedings of PVLDB '08*, pp. 230-45, August 23-28, 2008, Auckland, New Zealand
- Almeida, A., Castro, P., Menezes, T., Ramalho, G. (2003): Combining Idleness and Distance to Design Heuristic Agents for the Patrolling Task. In: *Proceedings of 2nd Brazilian Workshop on Games and Digital Entertainment*, Salvador, 2003.
- Almeida, A., Ramalho, G., Sanana, H., Tedesco, P., Menezes, T., Corruble, V., Chaveleyre, Y. (2004): Recent Advances on Multi-Agent Patrolling. In: Bazzan,

A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3161, pp. 464 – 483. Springer, Heidelberg.

Amigoni, F., Gatti, N., Ippedito, A. (2008). A Game-Theoretic Approach to Determining Efficient Patrolling Strategies for Mobile Robots. Proceedings of the 8th IEEE/WIC/ACM Conference on Intelligent Agent Technology (IAT); pages 500-503. Sydney, Austrália, 9-12 December 2008.

Anderson, S. (2009). Historical Dictionary of Terrorism, Third Edition, p. 28, The Scarecrow Press, 2009.

Andrade, M., Ferreira, C., Pereira, H. (2010). Uma Ontologia para a Gestão do Conhecimento no Processo de Desenvolvimento de Produto. Gest. Prod., São Carlos, v.17, n. 3, p. 537–551, 2010.

Araújo, M. T. (2010). Análise Comparativa de Técnicas de Patrulhamento com Terreno Heterogêneo. Trabalho de Conclusão de Curso. Centro de Informática, Universidade Federal de Pernambuco, Recife, PE.

Asada M. ; Kitano H., Noda I., & Veloso M. (1999). RoboCup : Today and tomorrow : what we have learned. In Artificial Intelligence 110, pp. 193-214, 1999.

Barber, K. S. e Martin, C. E. (1999). Agent Autonomy: Specification, Measurement, and Dynamic Adjustment. In Proceedings of the Autonomy Control Software Workshop at Autonomous Agents 1999 (Agents'99), 8-15, May, 1, 1999. Seattle, WA.

Basilico, N; Gatti, N., Amigoni, F. (2009a): Leader-Follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies. In Proceedings of the 8th ACM International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS); pages 57-64. Budapest, Hungary, May 10-15, 2009.

Basilico, N; Gatti, N., Amigoni, F. (2009b) Developing a Deterministic Patrolling Strategy for Security Agents. In Proceedings of the 9th IEEE/WIC/ACM Conference on Intelligent Agent Technology (IAT); pag. 565-572. Milan, Italy, September 15-18, 2009.

Basilico, N; Gatti, N.; Rossi, T; Ceppi, S; Amigoni, F. (2009c): Extending Algorithms for Mobile Robot Patrolling in the Presence of Adversaries to More Realistic Settings. In Proceedings of the 9th IEEE/WIC/ACM Conference on Intelligent Agent Technology (IAT); pag. 557-564. Milan, Italy, September 15-18, 2009.

Basilico, N; Gatti, N.; Villa, F. (2010): Asynchronous Multi-Robot Patrolling against Intrusion in Arbitrary Topologies. In Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI); pag. 1124-1229. Atlanta, USA, 11-15 July, 2010.

Batterman, R. (2006). Reduction. Encyclopedia of Philosophy, 2nd Edition. Ed.: Borchet, D. Detroit: Macmillan Reference.

Bayer, S., Damianos, L., Hirschman, L. & Strong, G. (2004). A Summary of Previous Grand Challenge Proposals for Cognitive Systems. The MITRE Corporation. Prepared for DARPA IPTO, September 2004 Version 1.4. Disponível

em http://www.mitre.org/work/tech_papers/tech_papers_05/05_0947/index.html.
Último acesso em 30 de janeiro de 2012.

BBC (2011). America's Day of Terror. Disponível em http://news.bbc.co.uk/1/hi/english/static/in_depth/americas/2001/day_of_terror. Último acesso em 26 de Agosto de 2011.

Bennet, J. (1988). Events and their Names. Hacket Publishing Co. Indianapolis.

Le Berre, D., Lynce, I. (2005): CSP2SAT4J: A simple CSP to SAT translator. In: Proceedings of the Second International Workshop on Constraint Propagation and Implementation, Volume II Solver Competition. (2005) 59–66.

Bittencourt, G. (2006). Inteligência Artificial: Ferramentas e Teorias. 3ª. Edição, Editora da UFSC.

Biolchini, J., Mian, P., Natali, A., Travassos, G. (2005) Systematic Review in Software Engineering. Technical Report ES 679/05. System Engineering and Computer Science Department, COPPE/UFRJ.

Birta, L. e Arbez, G. (2007). Modelling and Simulation: Exploring Dynamic System Behaviour. Springer-Verlag, London.

Booch, G., Rumbaugh, J., Jacobson, I. (2005) The Unified Modeling Language User Guide. 2nd Edition. Addison-Wesley Professional, 2005.

Bordini, R. e Hübner, J. (2009). Agent-Based Simulation Using BDI Programming in *Jason*. In: "Multi-Agent Systems Simulation and Applications", Eds: Adelinde M. Uhrmacher e Danny Weynspp, 451 – 476, CRC Press, Boca Raton, FL.

Boxwell, R. (1996). Vantagem Competitiva através do Benchmarking. Tradução José Carlos Barbosa dos Santos, São Paulo: Makron Books do Brasil, 1996. 255 pp.

Brigandt, Ingo and Love, Alan, "Reductionism in Biology", The Stanford Encyclopedia of Philosophy (Fall 2008 Edition), Edward N. Zalta (ed.), Disponível em <http://plato.stanford.edu/archives/fall2008/entries/reduction-biology>.

Bringsjord, S. E Schimanski, B. (2004), "Pulling it All Together" via Psychometric AI. AAAI Fall 2004 Symposium, October, 2004.

J. L. Burke, R. R. Murphy, D. R. Riddle, and T. Fincannon, "Task Performance Metrics in Human-Robot Interaction: Taking a Systems Approach," in *Performance Metrics for Intelligent Systems*, Gaithersburg, MD, 2004.

Cai, E., Gerding, P., McBurney, Niu, J., Parsons, S. e Phelps, S. (2009): Overview of CAT: A Market Design Competition. Version 2.0. Technical Report ULCS-09-005, Department of Computer Science, University of Liverpool, Liverpool, UK.

Casati, R. E e Varzi, A. (2008). Event Concepts. In: Understanding Events: From Perception to Action. Thomas F. Shipley and Jeff Zacks (eds.), New York: Oxford Press, 2008, pp. 31 – 54.

Casati, R. e Varzi, A. (2010). Events, The Stanford Encyclopedia of Philosophy, Spring, Edward N. Zalta (ed.), Disp. em

<http://plato.stanford.edu/archives/spr2010/entries/events/>. Último acesso em 31 de janeiro de 2012.

Cass, S. (2011). *Unthinking Machines: Artificial intelligence needs a reboot, say experts*. Disponível em <http://www.technologyreview.com/computing/36525/page1/>. Último acesso em 19 de junho de 2011.

Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W. Sheaffer, Sang-Ha Lee and Kevin Skadron. *Rodinia: A Benchmark Suite for Heterogeneous Computing*. In: *Proceedings of the IEEE International Symposium Workload Characterization*. Pp. 44-54, October 04 - 06, 2009.

Chevaleyre, Y. (2004): *Theoretical Analysis of the Multi-agent Patrolling Problem*. In: *Proc. of the Int. Conf. On Intelligent Agent Technology*, Beijing, China, pp. 302–308.

Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. *A Benchmark for 3D Mesh Segmentation*. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28(3), August 2009.

Chu, H., Glad, A., Simonin, O., Sempé, F., Drogoul, A., Charpillet, F. (2006). *Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation*. In: *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence - Vol.1*, 442 - 449 (ICTAI 2006).

Craighead, G., *High-Rise Security and Fire Life Safety*. In *Handbook of Loss Prevention and Crime Prevention*. Eds.: Fennelly, L., 4th Edition, Elsevier, pp. 370-375.

Davidson, D. (1987). *Essays on Actions and Events*. Oxford University Press.

De Palma, A. (2011). *City of Dust: Illness, Ignorance, Arrogance, and 9/11*. 1st. Ed. Pearson Education, Inc. New Jersey.

Drogoul, A., Landau, S., Muñoz, A. (2007). *RoboCup: Un aide pour l'AID?*. Unpublished paper.

Duchateau, F., Zohra Bellahsene, Ela Hunt. *XBenchMatch: a Benchmark for XML Schema Matching Tools*. In: *Proceedings of VLDB*, September, 2328, 2007, Vienna, Austria.

Elmaliach, Y., Agmon, N., Kaminka, G. (2006): *Multi-Robot Area Patrol under Frequency Constraints*. In: *Int. Conf. on Robotics and Automation*, Rome, Italy, pp. 385–390.

Elor, Y. e Bruckestein, A. (2009). *Efficiently Patrolling Hamiltonian and Two-Connected Graphs*. Technical Report CIS-2009-02. Faculty of Computer Science and the Goldstein UAV and Satellite Center.

Eytani, Y., Havelund, K., Stoller, S., e Ur, S. (2005). “*Toward a benchmark for multi-threaded testing tools*,” *Concurrency and Computation: Practice and Experience*, 2005.

Falcon, A. (2011). *Aristotle on Causality*, *The Stanford Encyclopedia of Philosophy* (Fall 2011 Edition), Edward N. Zalta (ed.), Disp. em

“<http://plato.stanford.edu/archives/fall2011/entries/aristotle-causality>”. Último acesso em 21 de setembro de 2011.

Fennely, L. (2004). *Handbook of Loss Prevention and Crime Prevention*, 4th Edition. Elsevier.

Feofiloff, P., Kohayakawa, Y., Wakabayashi, Y. (2009). Uma Introdução Sucinta à Teoria dos Grafos. Disp. em <http://www.ime.usp.br/~pf/teoriadosgrafos/>. Último acesso em 4 de outubro de 2011.

Ferber, J. (1999). *Multi-agent systems: An Introduction to Distributed Artificial Intelligence*. 1st. Eddition, Addison-Wesley.

Ferrara, A., Montanelli, S., Noessner, J., Stuckenschmidt, H. (2011). Benchmarking Matching Applications on the Semantic Web. Eds. Antoniou, G.; Grobelnik, M.; Simperl, E.; Parsia, B.; Plexousakis, D.; De Leenheer, P.; Pan, J., *Lecture Notes in Computer Science*, 2011, Volume 6644/2011, pp. 108-122, Springer Berlin / Heidelberg.

French, R. (2000). The Turing Test: The First Fifty Years. In: *Trends of Cognitive Science*, 4 (3), 2000, pp. 115 – 121.

Gatti, N. (2008). Game Theoretical Insights in Strategic Patrolling: Model and Algorithm in Normal-Form. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)*; pages 403-407. Patras, Greece, 21-25 July, 2008.

Glad, A., Simonin, O., Buffet, O., Charpillet, F. (2008). Theoretical Study of Ant-based Algorithms for Multi-Agent Patrolling. In: *Proceedings of the 18th European Conference on Artificial Intelligence*, pp. 626 - 630, IOS Press Amsterdam, Netherlands.

Glad, A., Simonin, O., Buffet, O., Charpillet, F. (2010). Influence of Different Execution Models on Patrolling Ant Behaviors: from Agents to Robots. In: *proc. of 9th Int. Conf. on Automous Agents and Multiagent Systems (AAMAS 2010)*. Eds.: van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10-14, 2010, Toronto, Canada, pp. 1173 – 1180.

Giot, Romain, Mohamad El-Abed, Christophe Rosenberger. GREYC Keystroke: a Benchmark for Keystroke Dynamics Biometric Systems. In *Proceedings of "IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, Washington : United States (2009).

Grinton, C. (1996). *A Testbed for Investigating Agent Effectiveness in a Multi-Agent Pursuit Game*. Bsc Thesis. The University of Melbourne. Australia.

Guardian (2011). Minsk metro bomb could be work of outsiders, says president. Disponível em <http://www.guardian.co.uk/world/2011/apr/12/minsk-metro-bomb-outsiders-president>. Último acesso em 26 de Agosto de 2011.

Guo, Y., Parker, L., Madhavan, R. (2006): Collaborative Robots for Infrastructure Security Applications. In: *Studies in Computational Intelligence (SCI)*, April 22, 2006, vol. 50, pp. 185 – 200. Springer, Heidelberg.

Guo, Y., Pan, Z., Heflin, J.: *An Evaluation of Knowledge Base Systems for Large*

OWL Datasets. In: Proc. of the 3rd Int. Semantic Web Conference (ISWC'04), Hiroshima, Japan (2004) 274–288.

Hanks, Pollack e Cohen (1993). Benchmarks, Test Beds, Controlled Experimentation, and the Design of Intelligent Agents. *AI Magazine*, Volume 14, pp. 17 – 42, Number 4 (1993).

Hayes, P., e Ford, K. (1995) Turing Test Considered Harmful. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence Montreal*, pp. 962 - 966.

J.P. Hespanha, Hyoun Jin Kim, and S. Sastry. Multiple-agent probabilistic pursuitevasion games. *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, 3:2432-2437, 1999.

Hoefer, Carl, (2012). Causal Determinism, *The Stanford Encyclopedia of Philosophy* (Spring 2010 Edition), Edward N. Zalta (ed.), Disponível em <http://plato.stanford.edu/archives/spr2010/entries/determinism-causal>. Último acesso em 29 de janeiro de 2012.

Howe, A., Dahman, E. (2002). A Critical Assessment of Benchmarking Comparison in Planning. In: *Journal of Artificial Intelligence Research*, 17, pp. 1 – 33.

Hsu, F., Jr. Hoane, A., Campbell, M., (2002). Deep Blue. In: *Journal of Artificial Intelligence - Chips challenging champions: games, computers and Artificial Intelligence*. Volume 134, Issue 1 - 2.

Huppler, K (2009). The Art of Building a Good Benchmark. In: *Performance Evaluation and Benchmarking, Lecture Notes in Computer Science*, Nambiar, Eds: R., Poess, M. ,Volume 5895. Springer-Verlag Berlin Heidelberg, 2009, p. 18 – 30.

Hwang, K., Lin, J., Huang, H. (2009): Cooperative Patrol Planning of Multi-Robot Systems by a Competitive Auction System. In: *International Joint Conference, Fukuoka, Japan, August*, pp. 18 - 21.

INSAG (1992). The Chernobyl accident: updating of INSAG-1: INSAG-6: a report by the International Nuclear Safety Advisory Group. — Vienna : International Atomic Energy Agency.

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. 1st Edittion. Wiley and Sons.

Jennings, N., Sycara, K., Wooldridge, M. (1998). A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems*, 1, pp. 275–306, Kluwer Academic Publishers, Boston.

John, L. K. (2006), Benchmarks, in: “Performance Evaluation and Benchmarking”. Eds. Lizy Kurian e Lieven Eeckhout, CRC Press, Boca Raton, FL.

Such, J.M., Alberola, J.M., Mulet, L., Espinosa, A., Garcia-Fornes, A., Botti, V.: Large-scale multiagent platform benchmarks. In: Dastani, M., El Fallah, Seghrouchni, A., Leite, J., Torroni, P. (eds.) *LADS 2007. LNCS*, vol. 5118, pp. 192–204. Springer, Heidelberg (2008).

_____. Trading Agent Competitions: The Ad Auctions Game for the 2010 Trading Agent Competition (Specification for Server Version 10.1.0.0). Disponível em http://aa.tradingagents.org/wp-content/uploads/2010/03/tac_aa_spec_2010-v10.1.0.01.pdf. Último acesso em 20 de novembro de 2011.

Kalyanakrishnan, K., Hester, T., Quinlan, M., Bentor, Y., e Stone, P. (2010). Three Humanoid Soccer Platforms: Comparison and Synthesis. J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 140 – 152, 2010. Springer-Verlag Berlin Heidelberg.

Kaneiwa, K., Iwazume, M., Fukuda, K (2006). An Upper Ontology for Event Classifications and Relations. In Proceedings of the 20th Australian joint conference on Advances in artificial intelligence, 394-403, Springer-Verlag, Berlin, Heidelberg.

Ketter, Wolfgang, Collins, John, Reddy, Prashant P. and Flath, Christoph M. (2011) The Power Trading Agent Competition (May 10, 2011). ERIM Report Series Reference No. ERS-2011-011-LIS. Disponível em <http://ssrn.com/abstract=1859866>.

Ketter, Wolfgang, Collins, John and Block, Carsten A., Smart Grid Economics: Policy Guidance Through Competitive Simulation (November 2010, 11). ERIM Report Series Reference No. ERS-2010-043-LIS. Disponível em <http://ssrn.com/abstract=1606913>.

Kitano, H.; Asada, M.; Kuniyoshi, Y; Noda, I; Osawa, E; Matsubara, H (1996): Robocup: A Challenge Problem for AI. In: AI Magazine, Vol. 18, Numero 1, pp 63-85.

Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., Susumu, Shimada, S. (1999). RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research, IEEE International Conference On Systems, Man, and Cybernetics 1999, pp. 639 - 646, IEEE Computer Society.

Kitchenham, B. (2004) Procedures for Performing Systematic Reviews. Keele University Technical Report TR/SE-0401, ISSN:1353-7776, July, 2004.

Klügl, F. (2009). SeSAm: Visual Programming and Participatory Simulation for Agent-Based Models. In: “Multi-Agent Systems Simulation and Applications”, Eds: Adelinde M. Uhrmacher e Danny Weynspp, 477 – 508, CRC Press, Boca Raton, FL.

Kosciansky, A. and Santos, M., 2007. Qualidade de Software, 2nd edition, Editora Novatec, Brasil.

Langle, G., Moen, R., Nolan, K., Nolan, T., Clifford, N., Provost. (2009). The Improvement Guide. A Practical Approach to Enhancing Organizational Performance. Second Edition. Jossey-Bass, Wiley Imprint, San Francisco, CA.

Lankford, W., Lankford. M. (2000). Benchmarking: Understanding the Basics. *The Coastal Business Journal*. v.1, nº 1.

Larman, C. (2000). Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. 1ª. Edição, Bookman Editora.

Lenat, D., (2008). The Voice of the Turtle: Whatever Happened to AI? AI Magazine, 2008, pp. 11-19.

Langley, G., Moen, R., Nolan, K., Nolan, T., Norman, C. (2009). The Improvement Guide: A Practical Approach to Enhancing Organizational Performance. John Wiley & Sons, Apr 20, 2009.

Levinson, R., Hsu, F., Schaeffer, J., Marsland, T., Wilkins, D. (1991). The Role of Chess in Artificial Intelligence Research. In: Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1, pp 546 – 552.

Less, R., Sultanik, E., Regli, W. (2009). Metrics for Multiagent Agents. In: Performance Evaluation and Benchmarking of Intelligent Systems, Eds: Madhavan et al., Springer+Business.

Lewis, D (1973). "Causation", Journal of Philosophy 70 (1973), pp. 556-567, reprinted in Philosophical Papers, vol. 2 (Oxford University Press: New York, 1986).

T.-Y. Liu, J. Xu, T. Qin, W.-Y. Xiong, and H. Li, "LETOR: Benchmark dataset for research on learning to rank for information retrieval," In: *SIGIR '07 Workshop on Learning to Rank for Information Retrieval*, 2007.

Lowther, A., Lindsay, B., (2009). Terrorism's Unanswered Questions. Edited by Adam B. and Beverly; Greenwood Publishing Group, Inc.

Machado, A., Ramalho, G., Zucker, J., Drogoul, A. (2003): Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures. In: Sichman, J.S., Bousquet, F., Davidsson, P. (eds.) MABS 2002. LNCS (LNAI), vol. 2581, pp. 155–160. Springer, Heidelberg.

Machado, A. (2002): Patrulha Multiagente: Uma Análise Empírica e Sistemática. M.Sc. Thesis, Centro de Informática, Univ. Federal de Pernambuco, Brasil.

Marino, A.; Parker, L.; Antonelli, G; Caccavale, F. (2009). Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach. In: Proc. of IEEE International Conference on Robotics and Automation.

Marino, A.; Parker, Caccavale, F; Parker, L; Antonelli, G; (2009). Fuzzy Behavioral Control for Multi-Robot Border Patrol. In: 17th Mediterranean Conference on Control & Automation, pp. 246-252, June, 2009.

Mazda, A. e Stone, P. (2006). A Multi-Robot System for Continuous Area Sweeping Tasks. In: proceedings of the IEEE International Conference on Robotics and Automation, pp. 1724-1729, Maio, 2009.

Menezes, T., Tedesco, P., Ramalho, G. (2006). Negotiator Agents for the Patrolling Task. In: Sichman, J.S., Coelho, H., Rezende, S.O. (eds.) IBERAMIA 2006 and SBIA 2006. LNCS (LNAI), vol. 4140, pp. 48–56. Springer, Heidelberg.

Menezes, T. R. (2006). Negociação em Sistemas Multiagente para Patrulhamento. M.Sc. Thesis, Centro de Informática, Univ. Federal de Pernambuco, Recife, Brasil.

Minsky, M. (1966). Computation: Finite and Infinite Machines. Prentice-Hall, p. 2.

Michel, F., Beurier, G., e Ferber, J. (2005). The TurtleKit simulation platform: Application to complex systems. In A. Akono, E. Tonyé, A. Dipanda, and K. Yétongnon, editors, Workshops Sessions, First International Conference on Signal & Image Technology and Internet-Based Systems SITIS' 05, pages 122–128. IEEE, november 2005.

Mitchel, T. (1997). Machine Learning, McGraw-Hill.

Moen, R., & Norman, C. (2006). Evolution of the PDCA Cycle. Disp. em http://www.xqmp.nl/content/bestanden/evolution_of_the_pdsa_cycle_jul06.pdf. Acessado em 11 de Janeiro de 2011.

Moreira, D., (2008). SimPatrol: Um Simulador de SMAs para Patrulhamento. Dissertação de Mestrado. Centro de Informática, Universidade Federal de Pernambuco, UFPE.

R. Narayanan, B. Ozisikyilmaz, J. Zambreno, G. Memik, A. Choudhary (2006). MineBench: A Benchmark Suite for Data Mining Workloads, In: IEEE International Symposium on In Workload Characterization, IEEE International Symposium on (2006), pp. 182-188.

R. Narayanan, B. Özisikyilmaz, J. Zambreno, G. Memik, and A. N. Choudhary. MineBench: A Benchmark Suite for Data Mining Workloads. In Proceedings of the IEEE International Symposium on Workload Characterization 2006, pages 182--188, 2006.

Newell, A. e Simon, H. (1966): Computer Science as Empirical Inquiry: Symbols and Search. In: Communications of the ACM. March, Vol. 19, Number 3, pp. 113 - 126.

Newborn, M. (2000). Deep Blue's contribution to AI. In: Annals of Mathematics and Artificial Intelligence. Science Publishers, 28, pp. 26 – 30.

Niu, J., Mmolohe, A., McBurney, P. e Parsons, S. (2009): CATP: A Communication Protocol for CAT Games Version 2.0. Technical Report ULCS-09-012, Department of Computer Science, University of Liverpool, Liverpool, UK.

Paruchuri, P; Tambe, M; Ordonez, F; Kraus, K (2006). Security in multiagent systems by policy randomization, Proceeding of AAMAS 06, Japan.

Paruchuri, P; Pearce, J; Tambe, M.; Ordonez, Fernando; Kraus, Sarit (2007). An Efficient Heuristic Approach for Security Against Multiple Adversaries. In: Proceeding of AAMAS 07, USA.

Paruchuri, P; Pearce, J; Mereck, J.; Tambe, M; Ordonez, F; Kraus, S. (2008) . Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games. In: Proc. of AAMAS, Padgham, Parkes, Mueller e Parsons (eds), Estoril, Maio, 12-16

Pelánek, R. BEEM: benchmarks for explicit model checkers, Proceedings of the 14th international SPIN conference on Model checking software, Pp. 263-267, Springer-Verlag Berlin, Heidelberg, 2007.

Pell, B. (1993). Strategy Generation and Evaluation for Meta-Game Playing. PhD Thesis. August, Trinity College, University of Cambridge, UK.

Pollack, M. e Riguette, M. (1990). Introducing the Tileworld: Experimentally Evaluating Agent Architectures, In: Proceedings of the Eighth National Conference on Artificial Intelligence , 1990 AAAI, pp. 183-189.

Prechelt, L. (1994). PROBEN1: A Set of Neural Network Benchmarking Rules. Technical Report 21/94. Universität Karlsruhe. September 30, 1994.

Pressman, R. (2010). Software Engineering: A Practioner's Approach. 7th ed. McGraw-Hill, New York.

Pronobis, A., Caputo, B., Jensfelt, P., and Christensen, H. 2009. A realistic benchmark for visual indoor place recognition. In: Journal of Robotics and Autonomous Systems. V. 58, n.1, pp. 81 – 96, January, 2010.

Purpura, P., (2006). Terrorisim and homeland security: an introduction with applications. Elsevier, Oxford.

Plunkett, L. (2011). World Champion Chess Program Banned For Cheating. Disponível em <http://gizmodo.com/5816993/world-champion-chess-program-banned-for-cheating>. Último acesso em 21 de julho de 2011.

Portugal, D. (2009). RoboCops: A Study of Coordination Algorithms for Autonomous Mobile Robots in Patrolling Missions. Msc. Dissertation, Faculty of Science and Technology, University of Coimbra, Portugal.

Portugal, D., Rocha, R. (2010). MSP Algorithm: Multi-Robot Patrolling based on Territory Allocation using Balanced Graph Partitioning. In: Proc. of Symposium on Applied Computin, Sierre, Switzerland, March, 22 – 26, 2010, pp. 1261 – 1266.

Portugal, D., Henggeler Antunes, C., Rocha, R. (2010): A Study of Genetic Algorithms for Approximating the Longest Path in Generic Graphs. In: Proc. 2010 IEEE Int. Conf. on Systems, Istanbul, Turkey, October 10-13.

T. Raths, J. Otten and C. Kreitz (2005). The ILTP Library: Benchmarking automated theorem provers for intuitionistic logic. In B. Beckert, editor, *Int. Conf. TABLEAUX-2005*, volume 3702 of *LNAI*, pages 333–337. Springer-Verlag, 2005.

Resnick, M. (1994). *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0-262-18162-2.

RoboCup (2011). <http://www.robocup2011.org>. Último acesso em 23 de agosto de 2011.

Robocup Rescue (2009). RoboCup Projects. Disponível em http://www.robocuprescue.org/wiki/index.php?title=Main_Page#Rescue_Simulation. Último acesso em 23 de janeiro de 2012.

Russel, S. e Norvig, P. (2004). Inteligência Artificial: Uma Abordagem Moderna. Tradução da 2^a. Edição, Editora Campus.

Sak, T., Wainer, J., Goldenstein, S. (2008). Probabilistic Multiagent Patrolling. In: Proc. of the Brazilian Symposium on Artificial Intelligence, G. Zaverucha, A. Loureiro (eds.): SBIA 2008, LNAI 5249, Springer-Verlag, Berlin – Heidelberg, pp. 124 – 133.

Sampaio, P. A., Ramalho, G., Tedesco, P. (2010). The Gravitational Strategy for the Timed Patrolling. 22nd IEEE International Conference on Tools with Artificial Intelligence. Arras, France, October 26-29 (2010).

Santana, H., Ramalho, G., Corruble, V., Ratitch, B. (2004): Multi-Agent Patrolling with Reinforcement Learning. In: Proc. of the Third Int. Joint Conf. on Autonomous Agents and Multiagent Systems, New York, vol. 3, pp. 1122–1129.

Sempé, F., Drogoul, A. (2003): Adaptive Patrol for a Group of Robots. In: Proc. of the Int. Conf. on Intelligent Robots and Systems, Las Vegas, Nevada.

Sim, S. (2003). A Theory of Benchmarking with Applications to Software Reverse Engineering. Doctorate Thesis. Graduate Department of Computer Science, University of Toronto.

Simon, H. e Newell, A. (1958) Heuristic problem solving: The next advance in operations research. *Operations Research*, 6, 1-10.

Spendolini, M. (1993). Benchmarking. Tradução: Kátia Aparecida Roque, São Paulo: Makron Books, 1993.

Shannon, R. (1975). *System Simulation: The Art and Science*. Prentice-Hall.

SICS (2011). TAC 2011 Participants. Disponível em <http://www.sics.se/tac/tacparts.php>. Último acesso em 23 de agosto de 2011.

Solingen, R., e Berghout, E. (1999). *The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development*. McGraw-Hill, 1st Edition, London, 1999.

Stone, P. (2003). Multiagent Competitions and Research: Lessons from RoboCup and TAC. In Gal A. Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup-2002: Robot Soccer World Cup VI*, Lecture Notes in Artificial Intelligence, pp. 224–237, Springer Verlag, Berlin, 2003.

Takahashi, T. (2009). RoboCup Rescue: Challenges and Lessons Learned. In: *Multiagent Systems: Simulation and Applications*. Eds: Uhrmacher, A. M., e Weyns, D. , pp.: 423-450, CRC Press.

Turing, A. M. (1950). Computing Machinery and Intelligence. In: *Mind*. Vol LIX., No. 236, October, 1950. pp 433 – 460.

Winther, R. (2009). Schaffner's Modelo of Theory Reduction: Critique and Reconstruction. In: *Philosophy of Science*, 66 (April, 2009), pp. 119 – 142.

van Beeck, E., Branche, C., Szpilman, D., Modell, J., e Bierens, J. (2005). A new definition of drowning: towards documentation and prevention of a global public health problem. *Bulletin of the World Health Organization*, 83 (11): 853-856, November, 2005.

Soratto, A., Varvakis, G. (2005). Uma metodologia para o gerenciamento da responsividade de serviços. In: Proc. of XII SIMPEP - Bauru, SP, Brasil, Novembro, 2005.

Vetsikas, I. e Selman, B. (2003). A Principled Study of the Design Tradeoffs for Autonomous Trading Agents. *AAMAS 2003 Proceedings of the II International Joint Conference on Autonomous Agents and Multiagent Systems*, Pages 473-480.

Weelman, M. e Wurman, P. (1999) A Trading Agent Competition for the Research Community. IJCAI – 1999. Workshop on Agent-Mediated Eletronic Commerce, Stockholm, 1999.

Weelman, Michael P. Wellman, Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Shou-de Lin, Daniel Reeves, William E. Walsh (2001). Designing the market game for a trading agent competition. IEEE Internet Computing. March/April, 2001.

Weng, P. (2008). What Makes JAGI Special. Journal of Artificial General Intelligence. 1. Disponível em http://journal.agi-network.org/portals/2/issues/JAGI_1_1-2.pdf. Published in 1, 2008-9-29, pp 1-2.

Wooldridge, M. (2009). An Introduction to Multiagent Systems. 2nd Edition. Willey and Sons.

Wolfs, Frank L. H (2004). Introduction to Scientific Method, Disponível em http://teacher.pas.rochester.edu/phy_labs/appendixe/appendixe.html#Heading9, Último acesso em 26 de junho de 2011.

Wörner, J; Wörn, H. (2006) Benchmarking of Multiagent Systems in a Production Planning and Control Environment. In: Multiagent Engineering: Theory and Applications in Enterprises. Eds.: Kirn, S., Herzog, O., Lockeman, P., Spaniol, O. Springer, Berlin-Heidelberg.

Zacks, J. e Tversky, B. (2001). Event Structure in Perception and Conception. In: Psychological Bulletin, Stanford University.

Zalta, E. (2011). Events. Stanford Encyclopedia of Philoshophy, <http://plato.stanford.edu/entries/events/>. Último acesso em 26 de março de 2011.

Zheng, Z. (1993). A Benchmark for Classifier Learning. Technical Report Number 474, Basser Department of Computer Science, The University of Sidney, November, 1993.

Zöller, A., Rothlauf, F., Paulussen, T., Heinzl, A. (2006). Benchmarking of Multiagent Systems. Eds. Kirn, S., Herzog, O., Lockman, P., Spaniol, O. Springer. pp. 557-574, Heidenberg.

Zuckerman, M. (2010). An American Crisis of Confidence. Disponível em <http://www.ihavenet.com/economy/An-American-Crisis-of-Confidence-MZ.html>. Último acesso em 15 de setembro de 2011.