

CSSA

Cadernos de Simulação Social e Ambiental

Volume 1, Número 2

Uso de Árvores de Decisão para Captura de Comportamentos de Agentes

*Josimara de A. Silveira
Thiago L. Paes
Diana F. Adamatti
Leonardo R. Emmendorfer*

Rio Grande, Dezembro de 2011

Cadernos de Simulação Social e Ambiental

Uma publicação online seriada, mantida pelo Grupo de Pesquisa em Simulação Social e Ambiental (GPSSA/FURG) do C3/FURG (Centro de Ciências Computacionais da Universidade Federal do Rio Grande).

Volume 1, Número 2
CSSA-02/2011
Dezembro/2011

Site: <http://cssa.c3.furg.br>

EDITORES

Antônio Carlos da Rocha Costa (C3/FURG)
Diana Francisca Adamatti (C3/FURG)
Graçaliz Pereira Dimuro (C3/FURG)
Marilton Sanchotene de Aguiar (CDTEC/UFPEL)

INSTRUÇÕES PARA OS AUTORES

Os Cadernos SSA publicam textos (artigos; relatórios de projetos de pesquisa; trabalhos de disciplinas; estudos individuais ou em grupos que tragam alguma contribuição sistemática, conceitual ou metodológica; monografias, etc.) relacionados com a área da Simulação Social e Ambiental.

Os textos para publicação nos Cadernos SSA devem ser formatados utilizando os modelos LaTeX ou Word da SBC (Sociedade Brasileira de Computação), disponíveis no site <http://www.sbc.org.br> (na seção de “Documentos”) ou diretamente no site dos Cadernos SSA.

Em princípio, não há limites máximo ou mínimo para o número de páginas.

Aceitam-se para publicação somente textos originais, não publicados anteriormente em nenhuma forma, impressa, digital ou eletrônica. Coletâneas de textos também são aceitas para publicação.

A submissão deve ser feita por email, para o endereço antoniocosta@furg.br, enviando pasta compactada em formato ZIP, contendo:

- arquivo fonte com o texto (em modelo LaTeX ou Word), conforme indicado acima;
- arquivo compilado em formato PDF, mostrando a forma final com que o texto deve ficar, ao ser publicado;
- arquivos em separado, com as figuras constantes no texto.

Os textos aceitos para publicação serão publicados com numeração serial, em regime de fluxo contínuo, e disponibilizados em formato PDF no site dos Cadernos SSA.

Uma vez publicado o texto nos Cadernos SSA, os autores comprometem-se, em todas as outras publicações que o texto vier a ter, a mencionar a publicação nos Cadernos como tendo sido a publicação original do mesmo.

Uso de Árvores de Decisão para a Captura de Comportamento de Agentes

Josimara de A. Silveira¹
Thiago L. Paes^{1,2}
Diana F. Adamatti¹
Leonardo R. Emmendorfer¹

¹Centro de Ciências Computacionais
Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 - 96201-900 - Rio Grande, RS, Brazil

²Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Avenida Ipiranga, 6681, Prédio 32 - 90619-900 – Porto Alegre – RS – Brazil

josi_silver@hotmail.com, {thiagopaes, dianaada, leonardo.emmendorfer}@gmail.com

Resumo. *Este trabalho apresenta a descrição de uma alternativa para detecção de padrões comportamentais de agentes. O principal propósito é analisar, através do estudo de cenários, com número diferente de agentes e de variáveis, o escopo de aplicação de árvores de decisão para a detecção comportamental de agentes.*

Abstract. *This work presents a new alternative to detect agents behavioral patterns. The main goal is to analyse, basing in scenarios with different quantities of agents and variables, the application scope of decision trees to detect these patterns.*

1. Introdução

Métodos de aprendizagem de máquina são adotados em muitas situações que demandam adaptação de comportamento, extração de conhecimentos, detecção de padrões, entre outras. Aprendizado de máquina está relacionado com a melhoria automática de performance na execução de uma tarefa, com base na experiência adquirida no passado. Esse trabalho baseia-se na aplicação do conceito de Árvores de Decisão (AD) que são classificadores que adotam uma estrutura de árvore condicional, onde cada nó especifica um teste a ser realizado em um único atributo [Duda et al. 2002, Mitchell 1997, Alpaydin 2010]. AD são ferramentas poderosas para classificação e previsão, pois fornecem regras que explicam o comportamento do sistema a ser modelado, além de fornecer um modelo de classificação.

Sistemas Multiagentes (SMA) é um sub-campo da inteligência artificial distribuída que se dedica ao estudo e modelagem de agentes autônomos, em um universo multiagente. Normalmente, cada agente pode ser descrito por (i) um conjunto de capacidades comportamentais, as quais definem a sua competência, (ii) um conjunto de objetivos e (iii) a autonomia necessária para utilizar suas capacidades e alcançar seus objetivos. Um agente é uma entidade computacional autônoma, que decide suas próprias ações [Sichman and Alvares 1997, Ferber 1999, Weiss 1999].

Aplicar AD no âmbito de SMA é uma abordagem promissora, alguns trabalhos discutem esse tipo de aplicação em domínios como robótica e cadeia de abastecimento [Stone and Veloso 1997, Li and Soh 2004].

Neste trabalho, regras são inferidas a partir do comportamento observado dos agentes da simulação e através dessas regras é possível entender o sistema de uma melhor forma. Quando se trata de AD existe uma limitação da técnica para a captura de relações complexas entre agentes e o meio em que estão inseridos. A ideia base do trabalho foi, em diferentes cenários multiagentes, descobrir até que ponto as AD podem ser utilizadas para a detecção da personalidade de agentes com alta competência.

2. Método Aplicado

O método utilizado consiste na geração de AD a partir dos dados de simulação de SMA com a finalidade de capturar comportamentos (programados ou não programados) dos agentes. O experimento foi aplicado em 3 cenários: presa-predador, fluxo granular e desastres, analisando 6 tipos de agentes: lobo, ovelha, pessoa, ambulância, bombeiro e policial. Em seguida o comportamento inferido pelas árvores foi analisado e comparado com o comportamento programado dos agentes.

A eficácia das árvores obtidas foi avaliada, assim como os bancos de dados responsáveis pela geração dessas árvores. Tal eficácia depende não só da complexidade do modelo mas também das peculiaridades do mesmo.

A figura 1 mostra as ferramentas utilizadas em cada uma das 3 etapas do desenvolvimento desse trabalho:

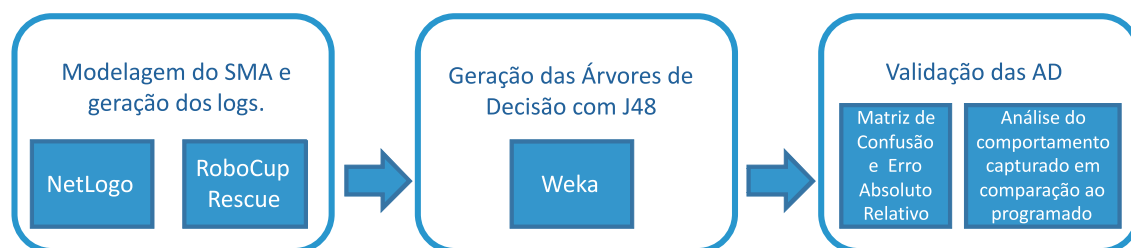


Figura 1. Etapas e ferramentas utilizadas no desenvolvimento do trabalho.

- Os SMA foram executados e o banco de dados foi gerado: foi utilizado o software NetLogo [Wilensky 1999] para os dois primeiros cenários (presa-predador e fluxo granular). Para o terceiro cenário foi utilizado o ambiente de simulação de desastres RoboCup Rescue [Morimoto 2001]. Com a finalidade de obter uma comparação justa entre as porcentagens de acerto na classificação da árvore de cada modelo, o número de instâncias contido em cada banco de dados é o mesmo, 13200 instâncias. Esse valor foi definido pelo terceiro cenário, a partir do produto dos 300 passos de simulação (valor padrão da competição RoboCup Rescue), 4 agentes de cada tipo e 11 variações em alguns parâmetros do modelo. Cada instância possui uma quantidade fixa de atributos para cada agente (12 no modelo presa-predador, 10 no fluxo granular e no simulador de desastres 33, 37 e 45 para os agentes do tipo policial, bombeiro e ambulância, respectivamente) e a ação tomada por ele.
- Geração das árvores de decisão a partir dos dados obtidos: foi utilizado o algoritmo J48 [Quinlan 1993] com auxílio do software WEKA [Hall et al. 2009]. Para cada agente foram geradas 2 árvores, uma com no mínimo 2 instâncias para cada folha e outra com no mínimo 200 instâncias para cada folha. Tal decisão foi tomada pois as primeiras árvores se apresentaram especializadas, tinham muitas folhas e poucas instâncias em cada folha.
- Validação das árvores obtidas: realizada por meio da análise das porcentagens de instâncias corretamente classificadas e erro absoluto relativo da classificação (taxas encontradas utilizando validação cruzada de 10 partes) além da análise da matriz de confusão de cada árvore.

3. Testes Realizados e Resultados Obtidos

Nesta seção analisaremos os resultados do método utilizado nos três cenários estudados, apresentando a porcentagem de instâncias corretamente classificadas, matriz de confusão e a árvore de decisão de cada agente.

3.1. Cenário presa-predador

O primeiro cenário trabalhado foi um ecossistema presa-predador, onde os agentes são do tipos lobo e ovelha. O ambiente é composto por uma grade toroidal de 30x30 casas sendo cada uma das casas (*patches*) representada pela presença ou não de grama. Os lobos interagem com as ovelhas em uma relação de predação, caçando-as. Quando o lobo consegue caçar uma ovelha sua energia aumenta. Tais agentes possuem uma visão em forma de cone com abertura voltada para o leste, comportamento adotado para elevar as chances de fuga dos agentes do tipo ovelha. Como o ângulo de abertura do cone foi definido em 200 graus os lobos enxergam quase tudo a sua volta, exceto a posição A s suas costas (O). A visão tem como característica o alcance de raio estipulado em dois *patches*, ou seja o lobo não observa sua presa A distância, somente tenta caçar quando já está próxima. Já as ovelhas sobrevivem fugindo dos lobos e alimentando-se de grama. A fuga das ovelhas está relacionada com a percepção ou não de lobos A sua volta. O método utilizado para fugir baseia-se em cada ovelha perceber a presença ou não de algum lobo e também a posição destes. Caso não existam lobos A sua volta, não há fuga e a ovelha movimenta-se aleatoriamente. Caso exista lobo, o

comportamento da ovelha é delimitado conforme regras de precedência. Quando em perigo, a ovelha olha primeiro para o noroeste (NO) e, caso não exista lobo, dirige-se para lá. Se existe lobo em NO, ela olha para o norte (N) e dirige-se até lá caso não haja predador, e assim por diante seguindo a ordem de procura: NO, N, NE, O, E, SO, S, SE. Tanto lobos quanto ovelhas perdem energia ao se mover.

Agente do tipo ovelha

Observando a primeira árvore de decisão dos agentes do tipo ovelha (Figura 2), com no mínimo duas instâncias por folha, podemos notar que quase todos as vizinhanças (somente a posição SO não apareceu na árvore), além de valores de energia, quantidade total de vizinhos e a ação de comer ou não, foram levados em conta na classificação, o que indica uma especialização nos ramos da árvore. Com o intuito de minimizar essa especialização, foi gerada uma segunda AD para os agentes do tipo Ovelha com no mínimo 200 instâncias para criação de cada folha. Nessa árvore (Figura 3) somente a classe “Aleatório” foi apresentada, pois nenhuma outra continha 200 instâncias para que fossem formadas outras folhas.

A segunda coluna da Tabela 1 mostra os valores da validação cruzada para a AD dos agentes do tipo ovelha com, no mínimo 2 instâncias por folha. Percebe-se que, embora a porcentagem de instâncias corretamente classificadas tenha sido de 93.75%, valor considerado alto para classificação por AD, o erro absoluto relativo também foi muito alto, chegando a quase 80%. Para explicar um erro tão alto, deve ser feita uma análise da matriz de confusão do modelo.

Tabela 1. Validação Cruzada - Agentes do tipo ovelha.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	93.75 % (12375)	93.7348 % (12373)
Instâncias incorretamente classificadas	6.25 % (825)	6.2652 % (827)
Erro relativo absoluto	79.8947 %	99.8198 %

Uma matriz de confusão, para ser considerada ótima, deveria contar com a concentração de valores em sua linha diagonal principal. Contudo, a Tabela 2 mostra que a maioria das instâncias diferentes de “Aleatório” foram classificadas erroneamente. O valor de 93,75% de instâncias corretamente classificadas é obtido somando as 85 instâncias classificadas corretamente como “NO”, as 12287 classificadas corretamente como “Aleatório” e as 3 classificadas corretamente como “N”, totalizando as 12375 instâncias corretamente classificadas como mostrado na Tabela 1. Isso se deve ao fato do banco de dados estar desbalanceado, existindo um número muito maior de movimentações do tipo “Aleatório” do que a escolha exata de uma das oito posições da vizinhança. A utilização do banco de dados desbalanceado se deu pois é essa a realidade desse cenário, onde na maioria das vezes o agente não tem a necessidade de escolher um lugar para se mover podendo fazer isso aleatoriamente. Nota-se que a matriz de confusão apresenta somente 4 das 9 opções possíveis de movimento do agente, isso ocorre pois as outras opções não foram escolhidas pelo agente já que ele “olha” primeiro para NO, N e NE.

Na terceira coluna da Tabela 1 estão os valores para validação cruzada da AD com 200 instâncias por folha, onde o erro relativo absoluto chegou a quase 100%, já que todas as instâncias que não eram da classe “Aleatório” foram classificadas como tal, pois essa era a única opção existente. A segunda matriz de confusão (Tabela 3) desse tipo de agente mostra a concentração de todas as instâncias na mesma coluna.

Tabela 2. Matriz de confusão - Agentes do tipo ovelha, com o mínimo de 2 instâncias/folha.

a	b	c	d	
85	694	0	0	a = NO
81	12287	5	0	b = Aleatório
2	39	3	0	c = N
0	4	0	0	d = NE

Agente do tipo lobo

A partir da primeira árvore de decisão dos agentes do tipo lobo (Figura 4) observa-se que, sempre que existe um agente do tipo ovelha na posição “Centro” o lobo captura-o, já que nesse caso os dois agentes encontram-se no mesmo

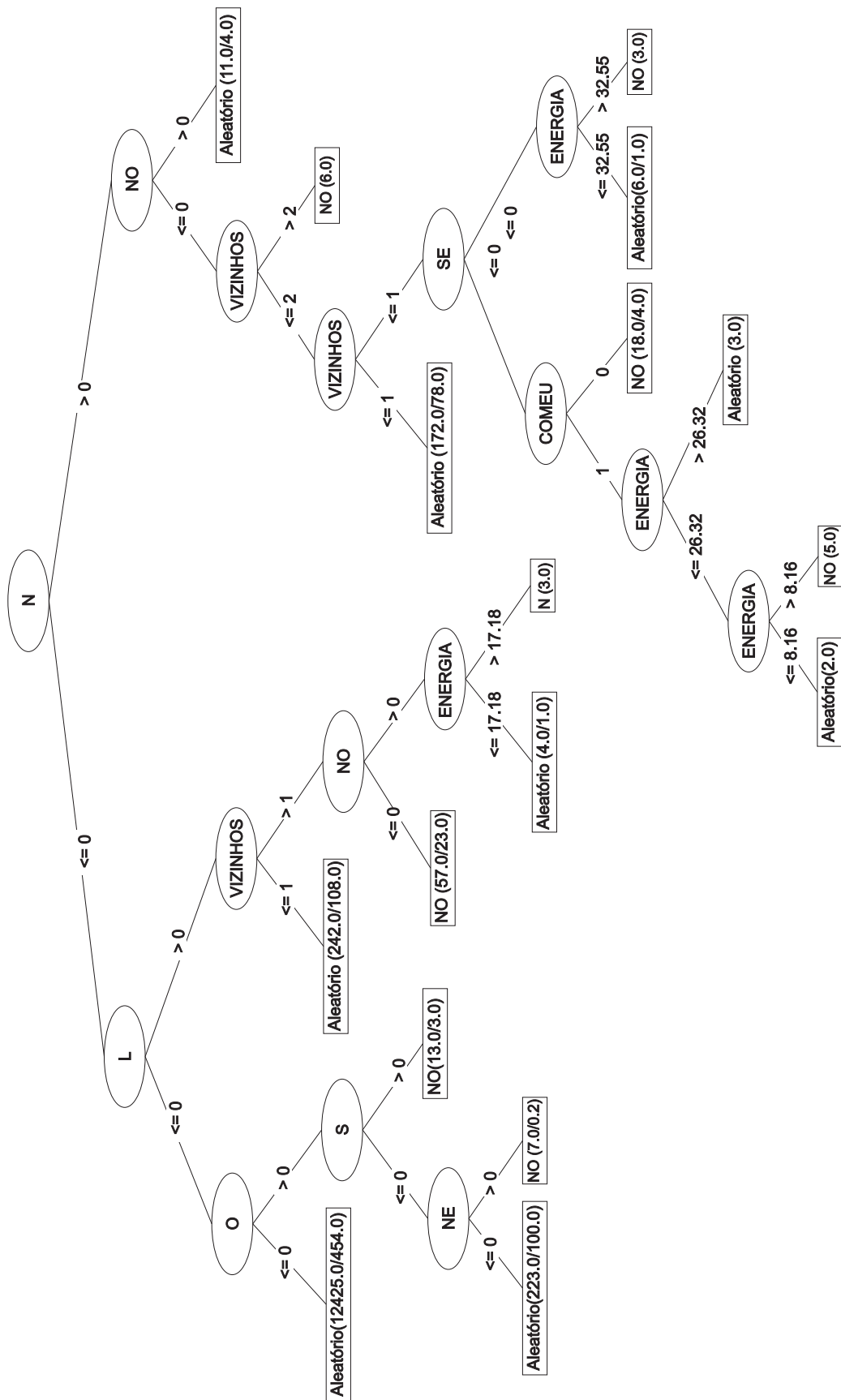


Figura 2. Árvore de decisão dos agentes tipo ovelha, com o mínimo de 2 instâncias/folha.

Aleatório(13200.827)

Figura 3. Árvore de decisão dos agentes tipo ovelha, com o mínimo de 200 instâncias/folha.

Tabela 3. Matriz de confusão - Agentes do tipo ovelha, com o mínimo de 200 instâncias/folha.

a	b	c	d	
0	779	0	0	a = NO
0	12373	0	0	b = Aleatório
0	44	0	0	c = N
0	4	0	0	d = NE

patch. Além disso, todos outros atributos presentes no banco de dados aparecem na árvore, com exceção de duas posições de vizinhança (NE e SO). Nota-se que, surpreendentemente, a posição Oeste (O), que o agente não enxerga, também aparece no banco de dados, dado o fato de que a existência de alguma ovelha naquela posição influencia o comportamento do sistema. Na segunda AD do tipo lobo (Figura 5), que utiliza 200 instâncias para a criação de uma folha, somente um atributo foi utilizado para bipartir os dados, a classificação mostrou um comportamento já identificado na primeira árvore (Figura 4) : sempre que há uma ovelha na posição “Centro” o lobo a captura.

A segunda coluna da Tabela 4 apresenta os dados da validação cruzada para a AD dos agentes do tipo lobo, com no mínimo 2 instâncias por folha, onde observa-se a existência de uma alta taxa de instâncias corretamente classificadas. Entretanto o erro relativo absoluto também é alto. O motivo do ocorrido é o mesmo citado na classificação dos dados dos agentes do tipo ovelha: existe um desbalanceamento nas classes, já que na maioria do tempo o lobo não consegue matar a ovelha. A matriz de confusão (Tabela 5) assim como no primeiro agente apresenta os valores concentrados em apenas uma coluna.

Tabela 4. Validação Cruzada - Agentes do tipo lobo.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	96.2576 % (12706)	96.1288 % (12689)
Instâncias incorretamente classificadas	3.7424 % (494)	3.8712 % (511)
Erro relativo absoluto	79.1235 %	94.7252 %

Novamente, para este tipo de agente, a AD com no mínimo 200 instâncias por folha não apresentou melhora no valor de erro relativo absoluto, como pode ser visto na terceira coluna da Tabela 4, o que reforça que o banco de dados desbalanceado tem alta influência na classificação realizada pela AD, já que mesmo com mais de 96% de instâncias corretamente classificadas o erro passou de 94%, o que é explicado na segunda matriz de confusão (Tabela 6), onde 510 de 539 (94.6%) são classificadas incorretamente na classe b = Matou.

3.2. Cenário 2: Fluxo Granular

O segundo cenário é um modelo de fluxo granular, o problema de evacuação de locais. A simulação conta com agentes de um único tipo, o tipo pessoa, e se apresenta numa grade de 21x21 *patches*. Os agentes têm apenas um objetivo: sair da sala. Todos nascem no mesmo lugar e se direcionam até a porta. A cada passo dado, existe o cálculo do ângulo de giro de cada agente, o qual simboliza para que direção ele vai se virar, e ficar de frente para a porta, para posteriormente dar um passo. Quando um agente quer se locomover até certo *patch* vizinho, ele verifica se o *patch* já contém mais de 3 pessoas ou se é um obstáculo (uma parede). Se não, ele pode ir até lá. Caso ele não consiga se movimentar, seu ângulo é randomizado e ele tenta se movimentar em outra direção qualquer. Cada vez que um agente consegue sair da sala, ele deixa de existir, é morto, e é criado um novo agente na posição onde todos nascem.

Agente do tipo pessoa

Observando a primeira AD para o agente do tipo pessoa (Figura 6), pode-se notar que, até o tempo 11, os agentes aproximam-se da porta, o que demonstra o espaço livre existente, onde os agentes podem se mover no destino desejado. Depois desse tempo a distância e quantidade de vizinhos começa influenciar no sucesso ao alcance do objetivo. Observa-se ainda que depois do tempo 12 e com uma distância da porta menor ou igual a 0.912031 os agentes aproximam da porta, o que pode demonstrar que o tumulto não ocorre tão próximo a saída e sim um pouco antes. Ainda é

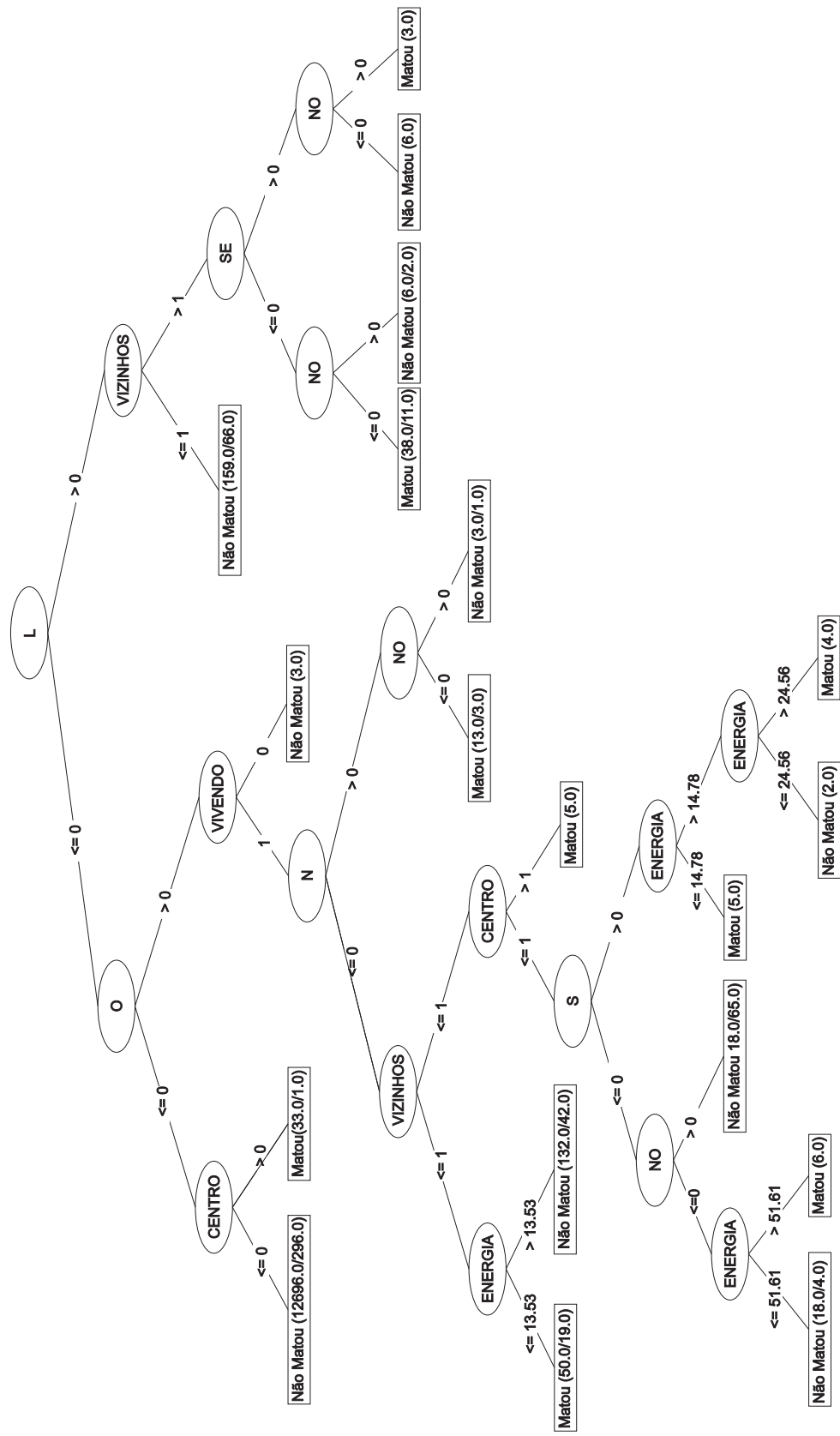


Figura 4. Árvore de decisão dos agentes tipo lobo, com o mínimo de 2 instâncias/folha.

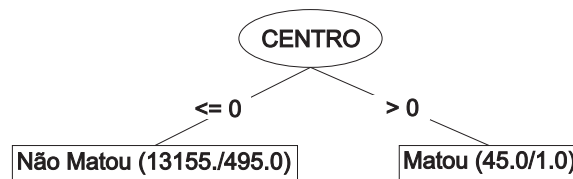


Figura 5. Árvore de decisão dos agentes tipo Lobo, com o mínimo de 200 instâncias/folha.

Tabela 5. Matriz de confusão - Agentes do tipo lobo, com o mínimo de 2 instâncias/folha.

a	b	
12617	44	a = Não Matou
450	89	b = Matou

possível notar que o atributo “Costas”, que indica agentes atrás do indivíduo em questão, não aparece na AD deixando claro que para esses agentes a presença de alguém ou não \tilde{A} suas costas não influencia na sua aproximação da saída, já que para isso ele precisa que haja espaço a sua frente e não atrás.

Observando a segunda AD (Figura 7) nota-se uma redução do número de atributos na modelagem do comportamento dos agentes do tipo pessoa. Agora só o tempo, a distância e a quantidade de agentes \tilde{A} frente foram necessários para a criação da árvore, sendo a distância o único atributo que aparece mais de uma vez. A AD obtida mostra-se mais enxuta e de fácil compreensão, sem deixar de capturar informações sobre o comportamento do sistema.

A segunda coluna da Tabela 7, apresenta os dados da validação cruzada para os agentes do tipo pessoa com, no mínimo 2 instâncias por folha. Pode-se perceber que 95.1439% (12559) das instâncias foram corretamente classificadas e o erro absoluto relativo foi de 23.2713%, valor consideravelmente menor que no primeiro cenário estudado, mesmo com o banco de dados deste cenário também desbalanceado (10471 para “Aproximou” e 2729 para “Afastou”).

Ao observar a matriz de confusão da primeira AD (Tabela 8), pode-se perceber que os valores mais altos estão localizados na diagonal principal, significando que a maioria das instâncias, tanto de uma classe quanto da outra, foram classificadas corretamente. O fato de apenas 4,01% (420 em 10471) das instâncias da classe “Aproximou” e 8,1% (221 em 2729) das instâncias da classe “Afastou” terem sido classificadas erroneamente é fator que proporciona um erro relativo absoluto de 23.2713%, bem menor que no primeiro cenário.

Para as AD com mínimo de 200 instâncias por folha, nota-se que os valores de classificação e erro relativo absoluto praticamente não se alteram (terceira coluna da Tabela 7), assim como a matriz de confusão (Tabela 9).

3.3. Cenário 3: Robocup Rescue

O terceiro cenário é uma simulação de ações de resgate em situações de desastres (terremotos, tsunamis, etc). O modelo possui 3 tipos de agentes: bombeiros, ambulâncias e policiais. Estes atuam sob vigilância de suas centrais, respectivamente: Estação de bombeiros, Centro de ambulâncias e Centro policial. Após os desastres, são considerados os colapsos entre os prédios, o acidente com civis e o espalhamento de incêndios. O desmoronamento de prédios sob as rodovias também é considerado e é responsável por grande parte da dificuldade imposta aos agentes, uma vez que, quando uma rua se encontra interditada por escombros, é necessário que os agentes (do tipo policial) retirem esses escombros para poder trafegar pela rua e assim salvar civis (ambulâncias) e/ou apagar incêndios (bombeiros). O foco do trabalho está neste cenário, o qual possui heurísticas elaboradas e um código maior e mais complexo.

Tabela 6. Matriz de confusão - Agentes do tipo lobo, com o mínimo de 200 instâncias/folha.

a	b	
12660	1	a = Não Matou
510	29	b = Matou

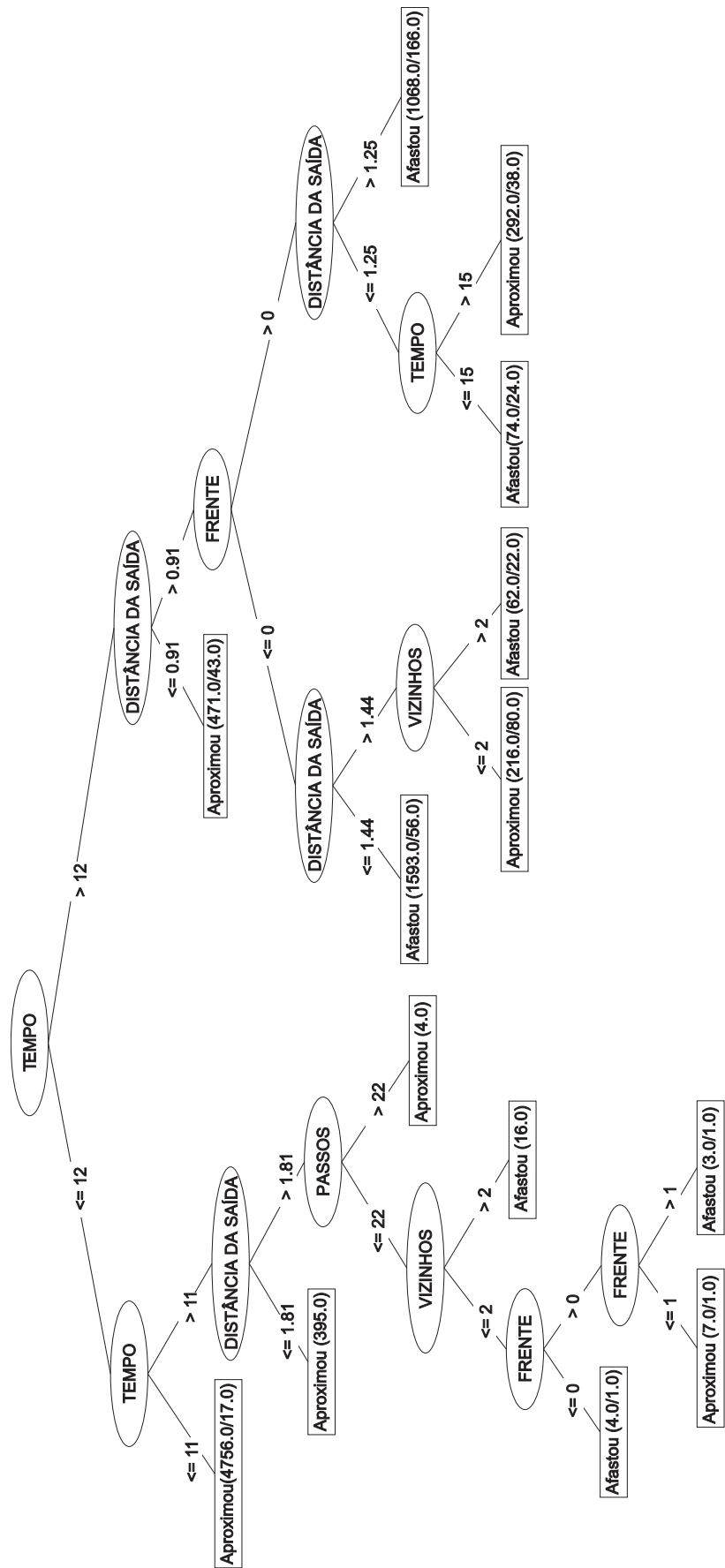


Figura 6. Árvore de decisão dos agentes tipo Pessoa, com o mínimo de 2 instâncias/folha.

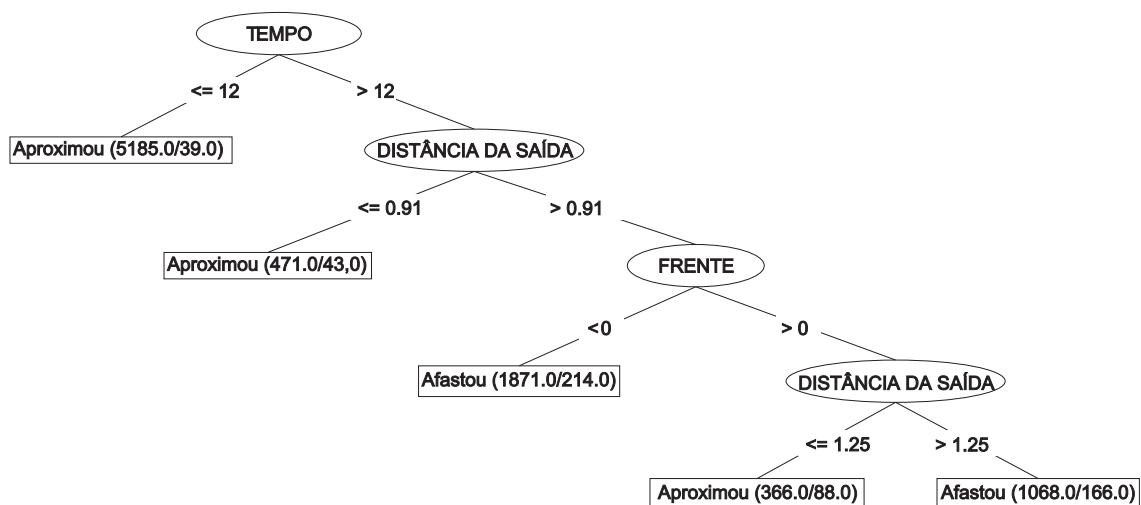


Figura 7. Árvore de decisão dos agentes tipo Pessoa, com o mínimo de 200 instâncias/folha.

Tabela 7. Validação Cruzada - Agentes do tipo pessoa.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	95.1439 % (12559)	95.1591 % (12561)
Instâncias incorretamente classificadas	4.8561 % (641)	4.8409 % (639)
Erro relativo absoluto	23.2713 %	23.657 %

Tabela 8. Matriz de confusão - Agentes do tipo pessoa, com o mínimo de 2 instâncias/folha.

a	b	
10051	420	a = Aproximou
221	2508	b = Afastou

Tabela 9. Matriz de confusão - Agentes do tipo pessoa, com o mínimo de 200 instâncias/folha.

a	b	
10054	417	a = Aproximou
222	2507	b = Afastou

Tabela 10. Estados/Ações dos agentes do tipo ambulância

Estado	Ação	Abreviatura
Não Comitado	Troca Tarefa	NC
Pegou Fogo, larga tarefa	Explorar Civil	PF
Tem tarefa e não sabe como ir	Explora Civil	TTNSCI
Tem tarefa e sabe como ir	Se movimenta em direção ao civil alvo	TTSCI
Não tem tarefa	Explora Civil	NTT
Está na tarefa	Age (larga, resgata ou carrega)	ENT
Está na tarefa, precisa ir ao refúgio e sabe ir	Se movimenta em direção ao refúgio	ENTPRSI
Está na tarefa, precisa ir ao refúgio e não sabe ir	Explora Refúgio	ENTPRNSI

Agente do tipo ambulância

Os objetivos de um agente do tipo ambulância são: resgatar civis que estejam soterrados por escombros e carregar civis feridos até hospitais. O método para tomada de ação dos agentes ambulância tem como base as heurísticas relacionadas à alocação de tarefas utilizando-se da comparação das capacidades do agente para efetuar a tarefa com a capacidade requerida pela tarefa. Além disso utiliza também probabilidade, onde a seleção da tarefa é feita por meio de um modelo inspirado em sociedades de insetos [FERREIRA JR. et al. 2010].

Para melhor entendimento, na Tabela 10 estão contidos os possíveis estado/ação dos agentes do tipo ambulância, juntamente com a abreviação.

Observando a primeira árvore de decisão dos agentes do tipo ambulância (Figura 8) podemos notar que a árvore utilizou-se de apenas 8 atributos (dentre os 45 presentes no banco de dados). A árvore inferiu que, quando está a 0 nós de distância do alvo, o agente do tipo ambulância deverá analisar o *damage* (Danos) da pessoa, e com base nesse valor e, eventualmente no valor da distância euclidiana até o alvo, decidir se troca de tarefa ou se de fato age. Quando o agente está localizado a mais de 0 nós do alvo, caso ele tenha uma tarefa ativa, a competência da tarefa deverá ser analisada. O valor da competência é menor do que 0 quando o agente não encontra o caminho até a tarefa selecionada, igual a 0 quando o alvo escolhido (dentre os possíveis) é o que está mais longe, maior do que 0 e menor do que 1 quando a tarefa selecionada não é a tarefa mais longe nem a mais próxima. É igual a 1 quando a distância euclidiana entre a tarefa selecionada e o local do agente é 0, ou seja, é a tarefa mais próxima.

A árvore demonstrou que se tarefa possuir competência >0.027348 , ou seja, não é o alvo que está colocado no lugar mais longe para se chegar, deve ser feita a ação de se movimentar até ele. Caso o alvo esteja em um dos locais mais longes alcançados pela percepção do agente, deve-se analisar o quão longe (em nós) o alvo está. Se estiver a menos de 59 nós do agente, a solução é a de efetuar a ação de mover-se para o refúgio, porém, se estiver a 57 nós ou menos de distância, os atributos coordenada y e coordenada x do agente devem ser levados em conta, com o intuito de descobrir se o melhor a fazer é mover-se para um refúgio alvo ou explorar outro refúgio. Ainda na primeira árvore nota-se que, caso o agente não possua uma tarefa ativa e se encontre longe de sua tarefa alvo, deve ser analisado, entre outros atributos, a distância euclidiana até o alvo e o fato do caminho até esse alvo estar bloqueado ou desbloqueado, assim variando a ação a ser tomada.

A segunda coluna da Tabela 11 apresenta os dados da validação cruzada para os agentes do tipo ambulância com, no mínimo, 2 instâncias por folha. Pode-se notar que a porcentagem de instâncias corretamente classificadas foi superior a 97% e que o erro relativo absoluto chegou a menos de 7%. A matriz de confusão (Tabela 12) mostra que os principais responsáveis pelo erro ter chego até 6.6721% foram as classes “f” e “g” os quais tiveram, respectivamente, 226 de 515 e 27 de 27 instâncias classificadas incorretamente.

A segunda árvore (Figura 9) apresentou a utilização de somente 4 atributos para definir qual a ação a ser tomada pelo agente, são eles: o tamanho do caminho, a competência da tarefa ativa, a competência do alvo e o tipo da localidade. Dentre os atributos listados, os três primeiros aparecem também na primeira árvore, já o tipo da localidade aparece pela primeira vez, e é analisado justamente testando se a localidade atual do agente é um refúgio ou não,

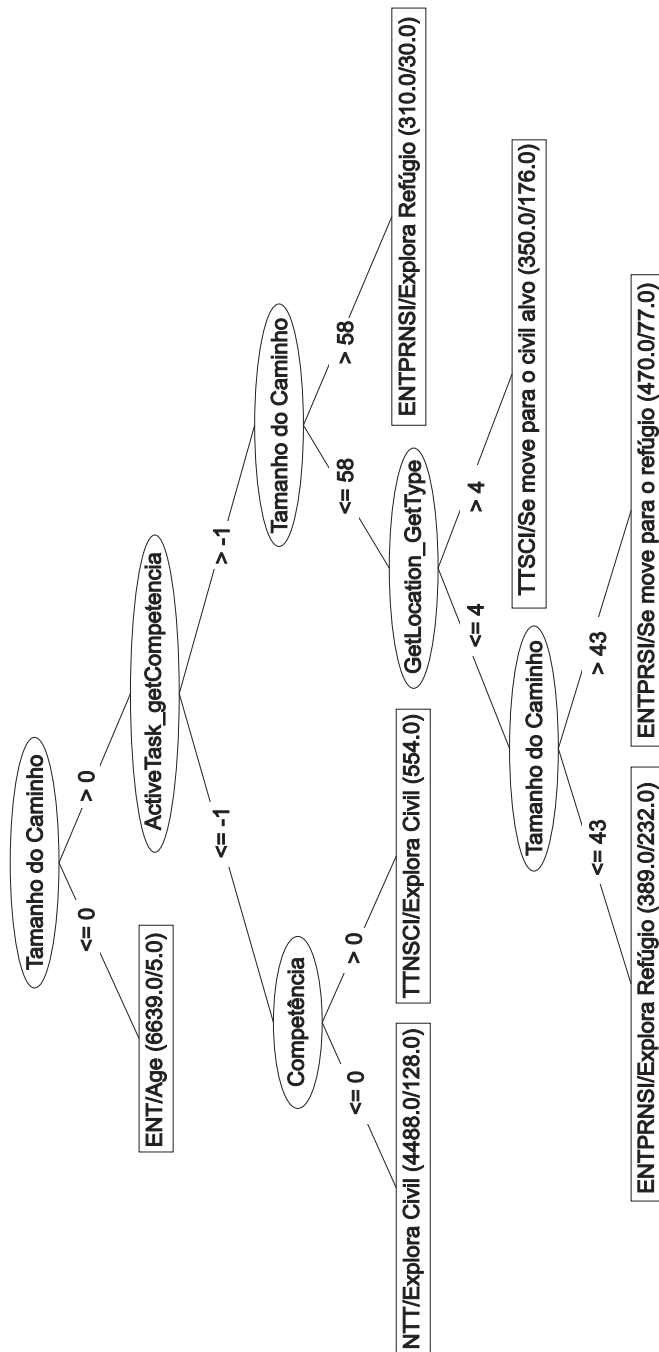


Figura 9. Árvore de decisão dos agentes tipo Ambulância, com o mínimo de 200 instâncias/folha.

interferindo diretamente na ação a ser tomada.

Para as AD com mínimo de 200 instâncias por folha, conforme apresenta a terceira coluna da Tabela 11, pode-se notar a pequena piora no número de instâncias corretamente classificadas (de 97.2424% para 95.0379%), juntamente com o aumento do erro relativo absoluto (de 6.6721% para 10.566%) devido a falhas de classificação como mostra a matriz de confusão (Tabela 13).

Tabela 11. Validação Cruzada - Agentes do tipo ambulância.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	97.2424 % (12836)	95.0379 % (12545)
Instâncias Incorretamente Classificadas	2.7576 % (364)	6.1364 % (655)
Erro relativo absoluto	6.6721 %	10.566 %

Tabela 12. Matriz de confusão - Agentes do tipo ambulância, com o mínimo de 2 instâncias/folha.

a	b	c	d	e	f	g	h	ESTADO/AÇÃO
4360	0	0	0	0	0	0	0	a = NTT/Explora Civil
12	643	0	0	0	0	0	0	b = TTNSCI/Explora Civil
0	0	195	0	82	0	0	0	c = TTSCI/Se move para o civil alvo
0	0	0	6633	0	0	0	1	d = ENT/Age
0	0	1	0	712	14	0	0	e = ENTPRSI/Se move para o refúgio
0	0	0	0	226	289	0	0	f = ENTPRNSI/Explora Refúgio
27	0	0	0	0	0	0	0	g = PF/Explora Civil
0	0	0	1	0	0	0	4	h = NC/Troca tarefa

Tabela 13. Matriz de confusão - Agentes do tipo ambulância, com o mínimo de 200 instâncias/folha.

a	b	c	d	e	f	g	h	ESTADO/AÇÃO
4360	0	0	0	0	0	0	0	a = NTT/Explora Civil
101	554	0	0	0	0	0	0	b = TTNSCI/Explora Civil
0	0	117	0	56	104	0	0	c = TTSCI/Se move para o civil alvo
0	0	0	6634	0	0	0	0	d = ENT/Age
0	0	131	0	454	142	0	0	e = ENTPRSI/Se move para o refúgio
0	0	4	0	85	426	0	0	f = ENTPRNSI/Explora Refúgio
27	0	0	0	0	0	0	0	g = PF/Explora Civil
0	0	0	5	0	0	0	0	h = NC/Troca tarefa

Agente do tipo bombeiro

O objetivo do agente tipo bombeiro é simples e intuitivo: apagar incêndios. Para melhor entendimento, na Tabela 14 estão contidas as possíveis estado/ação dos agentes do tipo Bombeiro, juntamente com a abreviação.

Ao gerarmos a primeira árvore de decisão (Figura 10) para os bombeiros, com no mínimo duas instâncias por folha, nos deparamos com um grave problema: demasiada especialização da árvore.

A árvore capturou várias características em relação aos agentes. Mostrou que, quando o agente está efetivamente junto à sua tarefa, ele atua nela. Caso não esteja, ele analisa se o bombeiro está ou não apagando algum fogo. Se o bombeiro possuir alguma tarefa ativa e possuir água em seu tanque, se movimentará em direção ao prédio alvo. Quando seu tanque está vazio o agente tem 3 possíveis decisões: (i) Mover-se para o refúgio alvo o qual sabe como chegar, quando a distância for menor ou igual a 63, (ii) Explorar por um refúgio aleatório, quando estiver longe de mais (64 nós ou mais), ou (iii) Movimentar-se em direção ao alvo, mesmo sem ter água para apagar o incêndio. Aqui vale a ressaltar o fato de, justamente em (iii), acontecer especialização da árvore (mais de 1000 instâncias classificadas de acordo com os diferentes ID's de alvos).

Tabela 14. Estados/Ações dos agentes do tipo Bombeiro

Estado	Ação	Abreviatura
Tem tarefa e não sabe como ir	Explora Incêndio	TTNSCI
Tem tarefa e sabe como ir	Se movimenta em direção ao incêndio alvo	TTSCI
Não tem tarefa	Explora Incêndio	NTT
Está na tarefa	Joga Água	ENT
Está na tarefa, não possui água em seu tanque e sabe ir até o refúgio	Se movimenta em direção ao refúgio	ENTSASI
Está na tarefa, não possui água em seu tanque e não sabe ir até o refúgio	Explora Refúgio	ENTSANSI

Quando o agente não estiver apagando fogo algum, se leva em conta sua distância do alvo, caso o agente já esteja sobre o alvo, ele explorará incêndio.

Se compararmos os resultados gerados sobre ambulâncias, bombeiros e policiais, chegaremos a conclusão de que em nenhuma outra árvore o atributo tempo mostrou-se importante para descrever o comportamento dos agentes, apenas aqui. É intuitivo que a medida em que tempo progride progridam também os incêndios, dificultando o trabalho dos bombeiros.

A árvore inferiu que, quando o agente bombeiro não está próximo do alvo, não possui tarefa ativa e o tempo não está muito avançado (menos que 206 espaços de tempo, de um total de 300) o agente pode tomar sua decisão analisando apenas um critério. Se o caminho até o alvo está livre o agente entra no estado/ação “ENTSANSI/Se Move para refúgio” senão, entra no estado/ação “TTNSCI/Explora Incêndio”. As duas folhas citadas acima são de fato um comportamento implícito no sistema capturado com sucesso pela árvore de decisão, se analisarmos com mais atenção chegaremos à conclusão de que a árvore conseguiu, curiosamente, reunir em uma só condição, uma ação relacionada à movimentação a procura por um prédio em chamas e outra relacionada à movimentação tendo como alvo chegar a um refúgio.

Por fim, quando o bombeiro não está próximo de seu alvo, não possui tarefa ativa e o tempo de simulação está muito avançado, a competência da tarefa selecionada deve ser analisada. Quanto mais próximo de 1 é o valor da competência, mais perto a tarefa está. Se a tarefa possuir uma competência menor ou igual a 0.619176, podemos dizer que ela está mais longe do que perto, e o bombeiro deverá explorar outros incêndios. Com uma competência maior que 0.619176, leva-se em conta a posição do bombeiro no eixo x e com base nesse valor, se definirá a ação do agente entre explorar refúgio ou explorar incêndios.

Fazendo uma análise da árvore do ponto de vista da validação cruzada, segunda coluna da Tabela 15, vemos que a porcentagem de instâncias corretamente classificadas chegou a um alto valor (98.0682%) e que o erro absoluto relativo foi de 4.8492%, demonstrando que a árvore de decisão conseguiu descrever com alta competência o comportamento dos bombeiros.

Através da Tabela 16, foi possível perceber que em apenas duas classes (“e” e “f”) existe um número considerável de instâncias classificadas incorretamente.

Na segunda AD para os agentes do tipo bombeiro (Figura 11) podemos notar que a maior parte dos atributos utilizados na primeira árvore, reaparece. Dentre os 8 atributos (desconsiderando a parte especializada) utilizados pela primeira árvore, 6 deles reaparecem na segunda árvore, 2 deles somem e 1 novo atributo (coordenada y) emerge.

Para a AD com mínimo de 200 instâncias por folha, conforme a terceira coluna da Tabela 15, há um aumento no número de instâncias incorretamente classificadas (de 1.9318% para 6.1364%), juntamente com o aumento do erro relativo absoluto (de 4.8492% para 12.6812%). Os motivos do aumento nos erros pode se explicar na matriz de confusão (Tabela 17), onde a classe “e” chega a possuir um número maior de instâncias incorretamente do que corretamente classificadas (239 incorretas contra 235 corretas).

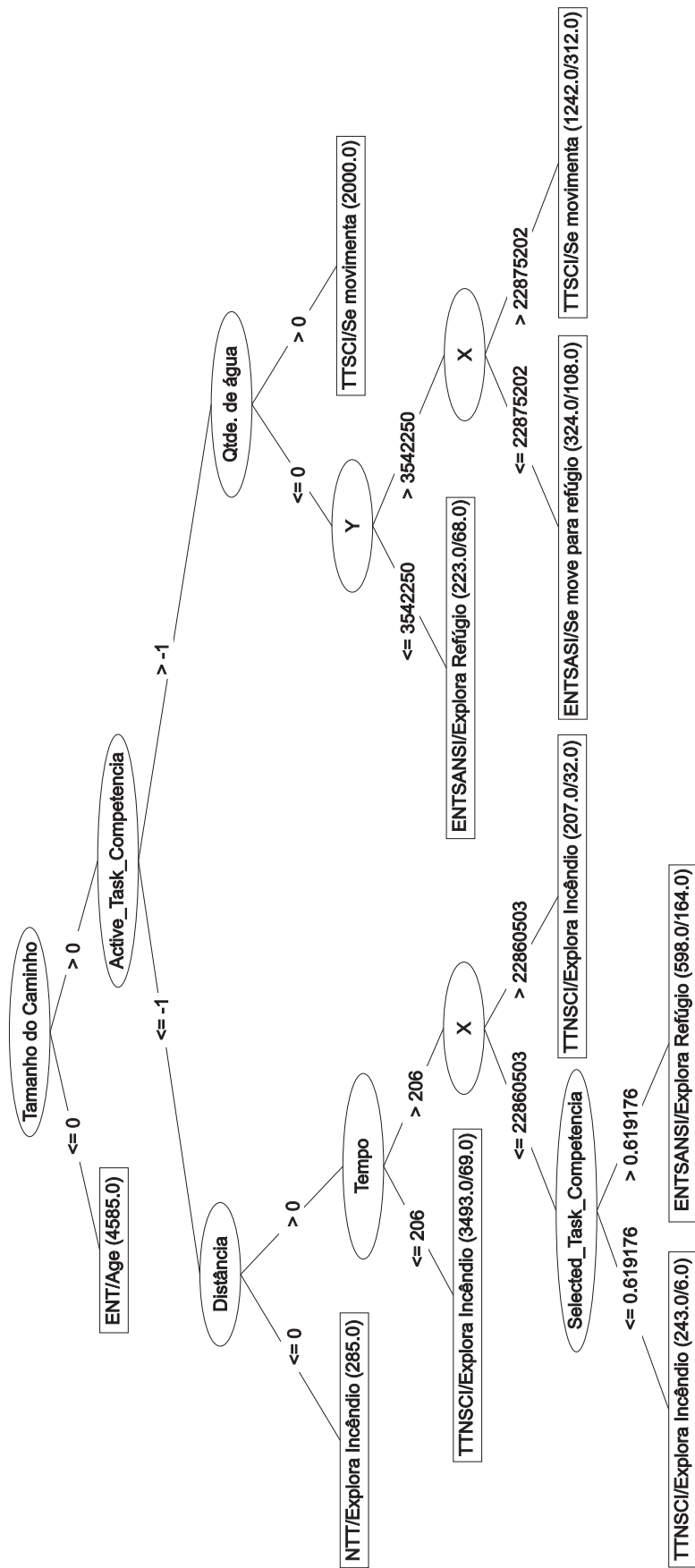


Figura 11. Árvore de decisão dos agentes tipo Bombeiro, com o mínimo de 200 instâncias/folha.

Tabela 15. Validação Cruzada - Agentes do tipo bombeiro.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	98.0682 % (12945)	93.8636 % (12390)
Instâncias Incorretamente Classificadas	1.9318 % (255)	6.1364 (810) %
Erro relativo absoluto	4.8492 %	12.6812 %

Tabela 16. Matriz de confusão - Agentes do tipo bombeiro, com o mínimo de 2 instâncias/folha.

a	b	c	d	e	f	ESTADO/AÇÃO
285	0	2	0	0	0	a = NTT/Explora Incêndio
0	3024	0	0	0	0	b = TTSCI/Se movimentada
0	0	3973	0	0	26	c = TTNSCI/Explora Incêndio
0	0	0	4585	0	0	d = ENT/Age
0	0	49	0	510	5	e = ENTSASI/Se move para refúgio
0	0	49	0	124	568	f = ENTSANSI/Explora Refúgio

Agente do tipo policial

O objetivo do agente tipo policial é o de retirar os escombros das ruas, assim deixando-as livres para poderem ser utilizadas. Para melhor entendimento das AD, na Tabela 18 estão contidas os possíveis estado/ação dos agentes do tipo policial, juntamente com a abreviação.

Ao gerarmos a primeira árvore (Figura 12) nos deparamos com o mesmo problema existente nos agentes do tipo bombeiro, especialização exagerada. Porém, ao analisar a árvore obtida, chega-se a conclusão de que a AD é apenas grande, e não complexa. Embora contenha um elevado número de folhas (1149), há um conjunto contendo 1138 de suas folhas agrupadas em um ramo, todos relacionando o Estado/Ação tomada com a rua da tarefa selecionada. Dentre essas 1138 folhas, apenas 4 se referem a Ação/Estado “TTNSCI/Explora Bloqueio” e o restante, refere-se ao Estado/Ação “NTT/Explora Bloqueio”. Vale ressaltar que o único ramo na árvore onde aparecem os Estado/Ação “NTT/Explora Bloqueio” são justamente as 1134 folhas já citadas. Entretanto as 4 folhas classificadas como “TTNSCI/Explora Bloqueio” reaparecem em outras partes da árvore.

A partir da árvore podemos prever algumas informações sobre o comportamento do agente. Primeiramente, caso ele esteja a uma distância de 0 nós do bloqueio alvo, ele simplesmente o retira. Caso a distância seja pequena (3 nós) ou menos, a ação que o agente toma pode ser de dois tipos: movimentar-se ou explorar bloqueio, e isso dependerá do caminho até o alvo estar desbloqueado ou não. Caso o caminho esteja bloqueado, o agente explorará bloqueio, caso contrário, se movimentará em direção ao alvo. Vale ressaltar aqui que as árvores de decisão capturaram um comportamento específico para os agentes com estratégia gulosa: quando perto do alvo, o agente com estratégia gulosa sempre sabe como chegar até o bloqueio, logo irá sempre se movimentar e não explorará bloqueios.

Caso o agente esteja longe do bloqueio alvo mais do que três nós, seu comportamento segue outras regras, levando em conta outras informações de sua percepção. A principal informação é sobre o caminho estar ou não desbloqueado. Com o caminho desbloqueado, o agente sabe como chegar a sua tarefa e se movimentar até ela. Caso contrário, outro atributo torna-se importante: o maior bloqueio alcançado pela percepção do agente. Embora esse atributo faça parte do banco de dados, ele não é utilizado pelo código fonte em momento algum, o que nos faz notar que a árvore foi

Tabela 17. Matriz de confusão - Agentes do tipo Bombeiro, com o mínimo de 200 instâncias/folha.

a	b	c	d	e	f	ESTADO/AÇÃO
285	0	2	0	0	0	a = NTT/Explora Incêndio
0	2878	0	0	112	34	b = TTSCI/Se movimentada
0	0	3875	0	0	124	c = TTNSCI/Explora Incêndio
0	0	0	4585	0	0	d = ENT/Age
0	226	45	0	235	58	e = ENTSASI/Se move para refúgio
0	60	97	0	52	532	f = ENTSANSI/Explora Refúgio

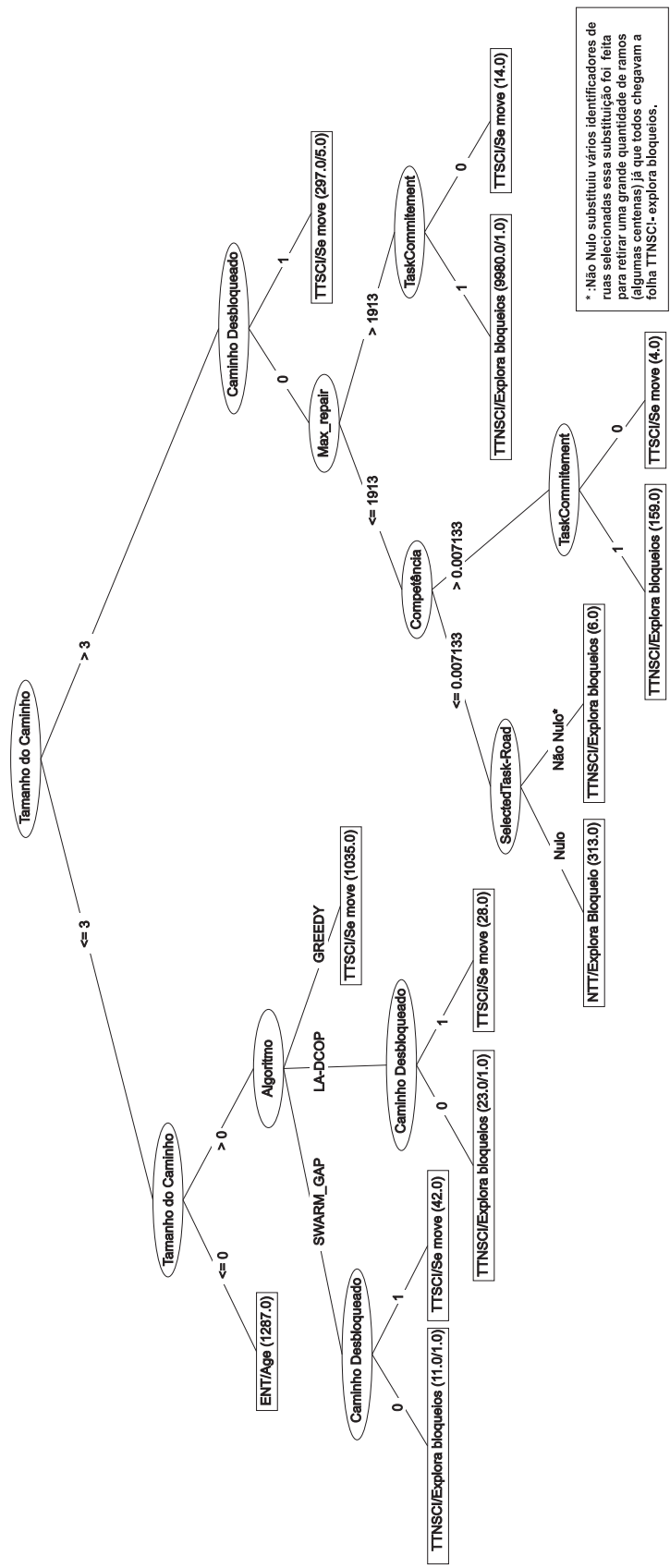


Figura 12. Árvore de decisão dos agentes tipo Policial, com o mínimo de 2 instâncias/folha.

Tabela 18. Estados/Ações dos agentes do tipo Policial

Estado	Ação	Abreviatura
Tem tarefa e não sabe como ir	Explora Bloqueio	TTNSCI
Tem tarefa e sabe como ir	Se Movimenta	TTSCI
Não tem tarefa	Explora Bloqueio	NTT
Está na tarefa	Retira Bloqueio (Age)	ENT

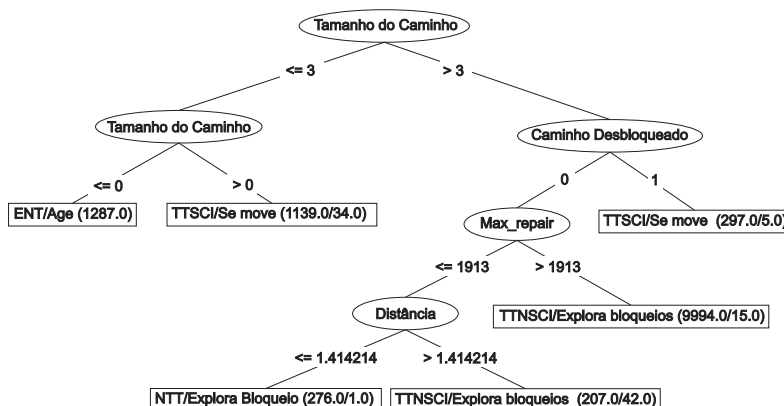


Figura 13. Árvore de decisão dos agentes tipo Policial, com o mínimo de 200 instâncias/folha.

capaz de capturar um novo conhecimento sobre o modelo e explicitá-lo.

Caso o maior bloqueio chegue a um valor maior do que 1913, é importante analisar a comitção do agente com tarefas e, como em nosso banco de dados apenas os agentes de estratégia gulosa não têm comitância, chega-se a conclusão de que, caso estejamos lidando com um agente guloso, este será capaz de largar a tarefa que tinha selecionado e movimentar-se em direção a uma nova tarefa, já se estivermos lidando com um agente de estratégia diferente, ele acabará por focar esforços na realização da mesma tarefa, mesmo que não saiba como chegar até ela e precise ficar explorando. Quando o maior bloqueio percebido pelo agente for menor do que 1913, o agente dá importância a competência da tarefa selecionada e analisa se ela é maior ou menor que 0.007133. Quanto mais próximo de 0 é o valor da competência, mais difícil do agente conseguir alcançar o alvo. Foi possível verificar que, normalmente, quando a competência é baixa, o agente não possui tarefa e assim sendo, como não sabe onde quer chegar, fica a explorar bloqueios. Caso ele saiba onde quer chegar (tem tarefa) e a competência se mostre próxima de 0, o agente acaba por não saber como chegar a tarefa e também explora bloqueios. A medida que a competência aumenta, leva-se em consideração a comitção de tarefas. Se o agente possui comitção (estratégias la-dcop e swarm-gap), ele explorará bloqueios. Se ele não possui comitção (gulosa), ele poderá sempre se movimentar em direção a sua tarefa selecionada, mesmo que para isso precise trocar de tarefa.

A segunda coluna da Tabela 19 apresenta os dados da validação cruzada para os agentes do tipo policial com, no mínimo, 2 instâncias por folha. Nota-se que tanto a porcentagem de instâncias corretamente classificadas quanto o erro absoluto relativo obtiveram valores extremamente satisfatórios, mostrando que mais de 99,8 % das instâncias foram corretamente classificadas e que o erro absoluto não chegou a 0,5%. Ao observar a matriz de confusão (Tabela 20), pode-se perceber que os valores mais altos estão localizados na diagonal principal, significando que a maioria das instâncias foram classificadas corretamente.

Na segunda AD gerada para estes agentes (Figura 13) podemos notar a efetiva generalização do comportamento, sendo que na primeira AD o comportamento era delineado dando importância a pequenos casos, como por exemplo o de um agente utilizando a estratégia swarm-gap que está próximo ao seu bloqueio alvo e ao mesmo tempo tem o caminho até ele bloqueado. Enquanto na árvore anterior, o comportamento do agente seria o de explorar o bloqueio, na nova árvore, levando-se em conta que a referida situação raramente acontece, a ação tomada seria a de movimentar-se mesmo com o caminho bloqueado. Alguns atributos contidos na primeira árvore desapareceram, fato proveniente da generalização da árvore. Em contrapartida, um fato chamou a atenção: o surgimento de um novo atributo a ser considerado para modelar o comportamento dos agentes, o atributo distância (euclidiana).

Para as AD com mínimo de 200 instâncias por folha, conforme a terceira coluna da Tabela 19, nota-se que o percentual de instâncias corretamente classificadas continua alto, diminuindo em menos de 1% (de 99.8788% para

98.9697%) em relação a primeira árvore e o erro relativo absoluto continua baixo, embora tenha uma maior variação (de 0.4809% para 4.3525%), o que pode ser observado pelas alterações na matriz de confusão (Tabela 21).

Tabela 19. Validação Cruzada - Agentes do tipo policial, com o mínimo de 2 instâncias/folha.

	2 inst/folha	200 inst/folha
Instâncias Corretamente Classificadas	99.8788 % (13184)	98.9697 % (13064)
Instâncias Incorretamente Classificadas	0.1212 % (16)	1.0303 % (136)
Erro relativo absoluto	0.4809 %	4.3525 %

Tabela 20. Matriz de Confusão - Agentes do tipo policial, com o mínimo de 2 instâncias/folha.

a	b	c	d	ESTADO/AÇÃO
309	4	0	8	a = NTT/Explora bloqueios
1	10173	0	3	b = TTNSCI/Explora bloqueios
0	0	1287	0	c = ENT/Retira bloqueio
0	0	0	1415	d = TTSCI/Se movimentata

Tabela 21. Matriz de confusão - Agentes do tipo Policial, com o mínimo de 200 instâncias/folha.

a	b	c	d	ESTADO/AÇÃO
253	61	0	7	a = NTT/Explora bloqueios
18	10127	0	32	b = TTNSCI/Explora bloqueios
0	0	1287	0	c = ENT/Retira bloqueio
1	17	0	1397	d = TTSCI/Se movimentata

4. Conclusão

A efetividade da detecção de padrões comportamentais de agentes a partir de árvores de decisão fica dependente não apenas da complexidade dos cenários de aplicação, mas também das peculiaridades específicas que cada cenário possui.

No caso do primeiro cenário, Presa-Predador, conclui-se que, mesmo sendo um SMA de baixa complexidade, o fato de ter-se um banco de dados desbalanceado gerou AD com alto grau de generalização e com erro relativo absoluto muito grande.

Já no segundo cenário, Fluxo Granular, onde também tem-se um SMA de baixa complexidade, as árvores capturaram com sucesso os comportamentos do agente “pessoa”, com alto percentual de instâncias classificadas corretamente e com erro relativo absoluto bem menor que o primeiro cenário.

O terceiro cenário, Robocup Rescue, onde tem-se um SMA mais complexo, pode-se perceber que um banco de dados balanceado tem maiores chances de modelar AD que capture o comportamento dos agentes com alta competência e baixo erro relativo absoluto.

Desta forma, pode-se concluir, baseado nos estudos realizados até o momento, que as AD podem capturar comportamentos em SMA mesmo em bancos de dados desbalanceados, como no segundo cenário, já que a classificação de cada instância (aproximar ou afastar do alvo) não era a ação tomada pelo agente e sim uma consequência de sua posição, vizinhança e movimentação. Também conclui-se que num cenário com uma complexidade maior, como o terceiro, as AD tem competência para capturar comportamento dos agentes já que, nesse caso além do banco de dados não se apresentar tão desbalanceado, possui muitos atributos o que favorece que uma AD se forme com as características mais marcantes para cada agente.

Referências

Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2nd edition. 537 p.

Duda, R., Hart, P., and Stork, D. (2002). *Pattern Classification*, chapter Chapter 8. Wiley Interscience, 2nd edition.

- Ferber, J. (1999). *Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, London. 509 p.
- FERREIRA JR., P. R., SANTOS, F., BAZZAN, A. L. C., EPSTEIN, D., and WASKOW, S. (2010). Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. In Springer, editor, *Auton Agent Multi-Agent Systems*, pages 421–443.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11(1).
- Li, X. and Soh, L. (2004). Applications of decision and utility theory in multi-agent systems. Technical Report 2004-0014, Computer Science and Engineering, Department of CSE Technical reports - University of Nebraska.
- Mitchell, T. (1997). *Machine Learning*. WCB McGraw-Hill.
- Morimoto, T. (2001). *How to Develop a RoboCupRescue Agent*, version 0 - 1st edition. RoboCupRescue Simulation System, edited by RoboCupRescue Technical Committee.
- Quinlan, J. (1993). C4.5: Programs for machine learning. Morgan Kaufmann Publishers.
- Sichman, J. S. and Alvares, L. O. (1997). Introdução aos sistemas multiagentes. In *Jornada De Atualização Em Informática*, pages 1–37, Brasília - UnB. Sociedade Brasileira de Computação.
- Stone, P. and Veloso, M. (1997). Using decision tree confidence factors for multiagent control. In *AAAI Technical Report WS-97-03*.
- Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press. 643p.
- Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling,, Northwestern University, Evanston, IL.