

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Milene Santos Teixeira

**MODELO SENSÍVEL À SITUAÇÃO PARA TOMADA DE DECISÃO
MULTIOBJETIVA EM AMBIENTES INTELIGENTES**

Santa Maria, RS
2018

Milene Santos Teixeira

**MODELO SENSÍVEL À SITUAÇÃO PARA TOMADA DE DECISÃO MULTIOBJETIVA EM
AMBIENTES INTELIGENTES**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

ORIENTADOR: Prof. Alencar Machado

COORIENTADOR: Prof. Michael Winter

Santa Maria, RS
2018

Teixeira, Milene
MODELO SENSÍVEL À SITUAÇÃO PARA TOMADA DE DECISÃO
MULTIOBJETIVA EM AMBIENTES INTELIGENTES / Milene
Teixeira.- 2018.
90 p.; 30 cm

Orientador: Alencar Machado
Coorientador: Michael Winter
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Ciência da Computação, RS, 2018

1. Multiobjetividade 2. Sensibilidade a Situação 3.
Sensibilidade ao Contexto 4. L-fuzzy 5. Ambientes
Inteligentes I. Machado, Alencar II. Winter, Michael
III. Título.

Sistema de geração automática de ficha catalográfica da UFSM. Dados fornecidos pelo autor(a). Sob supervisão da Direção da Divisão de Processos Técnicos da Biblioteca Central. Bibliotecária responsável Paula Schoenfeldt Patta CRB 10/1728.

©2018

Todos os direitos autorais reservados a Milene Santos Teixeira. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: milene.tsi@gmail.com

Milene Santos Teixeira

MODELO SENSÍVEL À SITUAÇÃO PARA TOMADA DE DECISÃO MULTIOBJETIVA EM AMBIENTES INTELIGENTES

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

Aprovado em 16 de março de 2018:

Alencar Machado, Dr. (UFSM)
(Presidente/Orientador)

José Palazzo Moreira de Oliveira, Dr. (UFRGS - Videoconferência)

Eduardo Kessler Piveta, Dr. (UFSM)

Santa Maria, RS
2018

DEDICATÓRIA

À todos aqueles que admiram a inteligência artificial ao invés de temê-la.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois mesmo que hoje eu esteja um pouco distante de seus caminhos, sei que sem Ele jamais teria chegado aqui. Aos meus pais (José e Liza), pois foram eles que me incentivaram a sempre estudar muito e com eles aprendi que minhas conquistas são resultados de meu esforço. Ao meu orientador Dr. Alencar Machado, por seus questionamentos que me possibilitaram abrir a mente e refletir sobre diversos pontos de meu trabalho. Agradeço a ele também por ter confiado em mim permitindo, diversas vezes, que eu buscasse meus próprios caminhos (como quando concordou com minha ida ao Canadá). Agradeço imensamente ao meu co-orientador Dr. Michael Winter, o qual prontamente buscou as informações necessárias para que eu participasse do programa de intercâmbio ELAP e realizasse parte de minha pesquisa ao seu lado. Agradeço por me receber de braços abertos e por sua imensa paciência ao repetir (diversas vezes) as explicações sobre \mathcal{L} -Fuzzy. Agradeço aos professores Dr. Eduardo Piveta e Dr. Daniel Lichtnow, participantes da banca do meu seminário de andamento, os quais contribuíram para que eu refletisse sobre pontos em aberto no meu trabalho. E por fim, agradeço a todos aqueles amigos que foram compreensivos nos momentos em que estive muito ocupada e não pude lhes dar atenção. A todos que acreditaram e torceram por meu sucesso, saibam que seu apoio foi fundamental! Muito obrigada!

O verdadeiro perigo não é que computadores começarão a pensar como homens, mas que homens começarão a pensar como computadores.

(Sydney J. Harris)

RESUMO

MODELO SENSÍVEL À SITUAÇÃO PARA TOMADA DE DECISÃO MULTIOBJETIVA EM AMBIENTES INTELIGENTES

AUTORA: Milene Santos Teixeira

ORIENTADOR: Alencar Machado

COORIENTADOR: Michael Winter

Cada dia mais surgem ambientes automatizados que buscam adicionar inteligência na tomada de decisão sobre os mais diversos problemas do mundo real, os quais, muitas vezes, apresentam objetivos múltiplos e conflitantes entre si. Em alguns casos, estes objetivos apresentam a mesma importância e, portanto, um ou outro não pode ser priorizado. Por lidarem com o raciocínio humano e questões cotidianas, Ambientes Inteligentes (Aml), frequentemente, apresentam esta característica. Este trabalho apresenta um modelo sensível à situação que busca auxiliar sistemas na tomada de decisão multiobjetiva para objetivos com mesma importância em Ambientes Inteligentes. Um sistema desenvolvido com base neste modelo identifica a situação de interesse a partir de dados do contexto obtidos de sensores existentes no ambiente; realiza a tomada de decisão multiobjetiva sem atribuir pesos aos objetivos; e executa automaticamente uma ação que visa controlar o ambiente. Para verificação do modelo proposto, foi desenvolvido um sistema utilizando como cenário um escritório em um Aml para a qual deseja-se prover conforto térmico ao usuário, evitando o consumo desnecessário de energia. Aliado a este sistema foi utilizado um dispositivo IoT que faz uso de sensores e é capaz de: a) obter valores de consumo e temperatura do ambiente e b) manipular a configuração de temperatura do ar condicionado. Como resultados, este trabalho disponibiliza uma biblioteca \mathcal{L} -fuzzy (utilizada no módulo de decisão) e mostra que a inclusão desta inteligência nos sistemas para Aml possibilita o alcance de ambos objetivos sem a necessidade de optar por um ou outro aspecto.

Palavras-chave: Multiobjetividade. Sensibilidade a Situação. Sensibilidade ao Contexto. \mathcal{L} -fuzzy. Ambientes Inteligentes.

ABSTRACT

CONTEXT SENSITIVE MODEL FOR MULTIOBJECTIVE DECISION MAKING IN AMBIENT INTELLIGENCE

AUTHOR: Milene Santos Teixeira
ADVISOR: Alencar Machado
CO-ADVISOR: Michael Winter

More and more there is the emergence of automated environments that seek to add intelligence to the decision making on the many different real world problems, which often have multiple and conflicting objectives. In some cases, these objectives have the same importance and, therefore, one or the other cannot be prioritized. For dealing with human reasoning and everyday activities, Ambient Intelligence (Aml) systems often present this feature. This paper presents a situation aware model that aims to assist systems in multi-objective decision making for objectives with same importance in Aml. From context data obtained from sensors in the environment, a system developed based on this model identifies the situation of interest; performs multiobjective decision making without assigning weights to the objectives; and automatically performs an action to control the environment. To verify the proposed model, a system was developed using as scenario an office in an Aml for which it is desired to provide thermal comfort to the user while avoiding unnecessary energy consumption. Allied to this system, it was used an IoT device that makes use of sensors and is able to: a) obtain data on consumption and temperature of the environment and b) manipulate the temperature setting of the air conditioner. As results, this work provides an \mathcal{L} -fuzzy library (used in the decision module) and shows that the inclusion of this intelligence in Aml systems allows the achievement of both objectives, without the need to choose one or the other aspect.

Keywords: Multiobjectivity. Situation Awareness. Context Awareness. \mathcal{L} -fuzzy. Ambient Intelligence.

LISTA DE FIGURAS

Figura 2.1 – Modelo conceitual do domínio proativo.	21
Figura 2.2 – SA e tomada de decisão.	28
Figura 2.3 – Representação gráfica de um conjunto <i>crisp</i> (a) X um conjunto fuzzy (b).	32
Figura 2.4 – Funções de pertinência para variável temperatura.	33
Figura 2.5 – Cota superior, inferior e ínfimo de M	36
Figura 2.6 – Representação de variáveis de pertinência Fuzzy X \mathcal{L} -Fuzzy.	38
Figura 2.7 – Modificadores deslocados com base em u	41
Figura 2.8 – Núcleo do controlador \mathcal{L} -fuzzy.	42
Figura 2.9 – Exemplo de núcleo do controlador.	43
Figura 2.10 – Exemplo de aplicação das regras de inferência.	43
Figura 2.11 – Defuzzificação usando operação ' α -cut'.	44
Figura 3.1 – <i>Framework</i> de controle do sistema multi-agentes.	52
Figura 4.1 – Inclusão no modelo conceitual dos conceitos e relações semânticas necessárias à solução de problemas multiobjetivos em Aml	56
Figura 4.2 – Modelo	57
Figura 4.3 – Algoritmo para desenvolvimento de sistemas sensíveis ao contexto para MoDM em Aml	61
Figura 4.4 – Diagrama de classes da biblioteca \mathcal{L} -Fuzzy.	62
Figura 5.1 – Aplicação do modelo no estudo de caso	66
Figura 5.2 – Tomada Inteligente	68
Figura 5.3 – Intervalo adequado para temperatura	71
Figura 5.4 – Intervalo adequado para consumo	72
Figura 6.1 – Amostra de registros utilizando a aplicação.	80
Figura 6.2 – Amostra de registros utilizando a aplicação, porém sem Situação de Interesse (S_i).	83

LISTA DE GRÁFICOS

Gráfico 5.1 – Gráfico variável de entrada: Temperatura.	74
Gráfico 5.2 – Gráfico variável de entrada: Consumo.	75
Gráfico 5.3 – Gráfico variável de saída: Configuração.	75
Gráfico 6.1 – Temperatura interna e externa com e sem uso da aplicação. No eixo x, D: dia, H:hora	81
Gráfico 6.2 – Consumo com e sem uso da aplicação. No eixo x, D: dia, H:hora	82

LISTA DE TABELAS

Tabela 5.1 – Valores mínimo, máximo e ótimo para variáveis de entrada e saída.....	69
Tabela 5.2 – Temperatura – Intervalo adequado para conforto térmico.....	71
Tabela 5.3 – Consumo – Intervalo adequado para eficiência energética.....	71
Tabela 5.4 – Variáveis linguísticas de entrada e saída.....	73
Tabela 5.5 – Valores obtidos na saída do LFC de acordo com valores de entrada.....	77
Tabela 6.1 – Temperatura média (externa) por tipo de teste. a) utilizando aplicação, b) não utilizando aplicação, c) utilizando aplicação sem detectar a situação de interesse (<i>Si</i>)	79
Tabela 6.2 – Dados da execução dos testes. a) utilizando aplicação, b) não utilizando aplicação, c) utilizando aplicação sem <i>Si</i>	80
Tabela 6.3 – Resultados dos testes: valores médios conforme cenário	83

LISTA DE QUADROS

Quadro 5.1 – Representação do evento ev_1	72
Quadro 5.2 – Representação do evento ev_2	73
Quadro 5.3 – Relações \mathcal{L} -fuzzy correspondentes às variáveis linguísticas.	74
Quadro 5.4 – Regras de inferência.	76
Quadro 5.5 – Representação do evento ev_3	77

LISTA DE ABREVIATURAS E SIGLAS

<i>Aa</i>	Ação automatizada
<i>AmI</i>	Ambientes Inteligentes
<i>ANN</i>	<i>Artificial Neural Network</i>
<i>ANP</i>	<i>Analytic Network Process</i>
<i>COG</i>	<i>Center Of Gravity</i>
<i>E</i>	Evento
<i>ev</i>	Evento
<i>evSet</i>	Conjunto de eventos
<i>EMO</i>	<i>Evolutionary Multi-objective Optimization</i>
<i>Eo</i>	Entidade objeto
<i>Es</i>	Entidade sujeito
<i>ES</i>	Sistemas Especialistas
<i>FC</i>	Controlador Fuzzy
<i>FCM</i>	<i>Fuzzy Cognitive Maps; Fuzzy Consensus Model</i>
<i>Fe</i>	Evento de fim
<i>FMCDM</i>	<i>Fuzzy Multi-criteria Decision Making</i>
<i>FOM</i>	<i>First of Maxima</i>
<i>GDM</i>	<i>Group Decision Making</i>
<i>I</i>	Entidades de entrada
<i>Ie</i>	Evento de início
<i>IoT</i>	<i>Internet of Things</i>
<i>LFC</i>	Controlador \mathcal{L} -fuzzy
<i>MeOM</i>	<i>Mean of Maxima</i>
<i>MOGA</i>	<i>Evolutionary Multi-objective Genetic Algorithm</i>
<i>MOPSO</i>	Multi-objective Particle Swarm Optimization
<i>O</i>	Entidades de saída
<i>obj</i>	Objetivo

<i>P</i>	Problema multiobjetivo
<i>RCOM</i>	<i>Random Choice of Maxima</i>
<i>SA</i>	<i>Situation Awareness</i>
<i>MCDM</i>	<i>Multiple Criteria Decision Making</i>
<i>MADA</i>	<i>Multiattribute Decision Analysis</i>
<i>MADM</i>	<i>Multiple Attribute Decision Making</i>
<i>MODM</i>	<i>Multiple Objective Decision Making</i>
<i>p</i>	Predicado; Padrão
<i>PSO</i>	<i>Particle Swarm Optimization</i>
<i>R</i>	Relações semânticas; Regras de inferência
<i>S_a</i>	Situação atual
<i>S_e</i>	Serviço
<i>S_i</i>	Situação de interesse
<i>S_p</i>	Situação pré-condição
<i>TOPSIS</i>	<i>Technique for Order of Preference by Similarity to Ideal Solution</i>
<i>v</i>	Variável

LISTA DE SÍMBOLOS

\mathcal{L}	Lattice
μ	Grau de pertinência
\leq	Menor ou igual
\in	Pertence
\notin	Não pertence
\geq	Maior ou igual
\cup	União
\cap	Conjunção
\curvearrowright ou $'$	Conversão
\circ ou $;$	Composição
$L - rel$	Relação \mathcal{L} -fuzzy
$L - set$	Conjunto \mathcal{L} -fuzzy
L^{in}	L-rel com valores de entrada
L^{out}	L-rel com valores de saída
\subseteq	Inclusão
\circ	Grau
$\alpha - cut$	Operação de defuzzificação
$HeytA$	Heyting algebra
W	Watts
\vee	Operador lógico 'ou'
\wedge	Operador lógico 'e'

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS E CONTRIBUIÇÕES	19
1.2	ORGANIZAÇÃO DO TEXTO	20
2	REFERENCIAL TEÓRICO	21
2.1	CONSCIÊNCIA SITUACIONAL	21
2.1.1	Sensibilidade ao contexto e situação	25
2.2	TOMADA DE DECISÃO MULTICRITÉRIOS	28
2.3	LÓGICA FUZZY	31
2.3.1	Etapas do processamento	32
2.3.1.1	<i>Fuzzificação</i>	32
2.3.1.2	<i>Processo de inferência fuzzy</i>	33
2.3.1.3	<i>Defuzzificação</i>	34
2.3.2	Considerações	34
2.4	TEORIA DOS LATTICES E \mathcal{L} -FUZZY	35
2.4.1	Conjuntos e Relações \mathcal{L}-Fuzzy	37
2.4.1.1	<i>Controlador \mathcal{L}-fuzzy</i>	38
2.4.1.1.1	Variáveis linguísticas	39
2.4.1.1.2	Fuzzificação	41
2.4.1.1.3	Base de regras	41
2.4.1.1.4	Módulo de Inferência e Defuzzificação	42
2.5	AÇÃO AUTOMATIZADA	45
3	TRABALHOS RELACIONADOS	47
3.1	SENSIBILIDADE AO CONTEXTO E/OU SITUAÇÃO EM AMI	47
3.2	<i>SITUATION AWARENESS</i> E LÓGICA FUZZY	48
3.3	TOMADA DE DECISÃO MULTI OBJETIVA FUZZY	50
3.4	PROBLEMA MULTI OBJETIVO: CONFORTO X ENERGIA	51
3.5	CONSIDERAÇÕES	53
4	MODELO PROPOSTO	55
4.1	TOMADA DE DECISÃO	61
5	APLICAÇÃO DO MODELO	65
5.1	ESTUDO DE CASO: CONTROLE DE CONFORTO TÉRMICO X EFICIÊN- CIA ENERGÉTICA	67
5.1.1	Problema Multiobjetivo	69
5.1.2	Contexto de Interesse	69
5.1.3	Situação de Interesse	70
5.1.4	Módulo de decisão	73
5.1.5	Ação automatizada	77
6	RESULTADOS	79
6.1	CONSIDERAÇÕES	83
7	CONCLUSÃO	85
	REFERÊNCIAS BIBLIOGRÁFICAS	87

1 INTRODUÇÃO

Ambientes inteligentes (Aml) (AARTS; WICHERT, 2009) é uma área interdisciplinar que visa adicionar inteligência em ambientes tornando-os sensíveis e adaptáveis às necessidades dos usuários. De acordo com (AUGUSTO; MCCULLAGH, 2007), o alcance desta inteligência está sujeito a dois fatores: a) as tecnologias implementadas (sensores e dispositivos interconectados por redes) e b) a inteligência do *software* utilizado para tomada de decisão. Normalmente, as tecnologias utilizadas em sistemas para Aml são tecnologias que visam atender as necessidades do usuário agindo de uma forma responsiva e não intrusiva. Além disso, alguns destes sistemas buscam tomar decisões sobre o mundo real, seja com o objetivo de identificar a maneira mais adequada para evitar ou resolver um problema, ou simplesmente para fazer uma escolha em um conjunto de alternativas.

Quando falamos em escolhas e problemas do mundo real, é comum que estes envolvam múltiplos objetivos e que, muitas vezes, estes objetivos sejam conflitantes. Isto ocorre nas mais diversas áreas como eficiência energética, engenharia, economia e até nas situações mais simples do dia a dia. Em um exemplo mais específico, supondo que se busque adquirir um eletrodoméstico que possua ambas qualidades: baixo consumo energético e alta eficiência, com a lógica clássica, seria necessário identificar ambos aspectos separadamente e escolher o de melhor valor, por exemplo. Entretanto, por que baixo consumo deveria ter prioridade sobre o outro aspecto ou vice-versa? Quando ambos aspectos são relevantes, um não pode (e não deve) ser priorizado. Neste contexto, surge a teoria da otimização multiobjetiva (FANDEL; GÁL; HANNE, 1997). Multiobjetividade corresponde a um problema para o qual devem ser alcançados diversos objetivos ao invés de apenas um e, em geral, estes objetivos são conflitantes entre si não existindo uma solução que seja considerada ideal para todos.

Normalmente, problemas de otimização multiobjetiva são solucionados por escalabilidade (MIETTINEN, 2012), ou seja, o problema é conferido em um único ou em uma família de problemas de otimização com objetivo único. Após esta conversão, podem ser aplicados métodos de otimização para problemas com objetivo único. Entretanto, devido a aspectos como diferença de unidades, características conflitantes, etc., nem sempre é possível encontrar uma única solução adequada com relação a todos objetivos apresentados. Além disso, muitos destes problemas não são lineares e, portanto, são necessários novos conceitos, métodos e maneiras de pensar que os manipulem eficientemente, pois métodos existentes para otimização de objetivo único não são suficientes.

Em Aml problemas multiobjetivos são frequentes. Se analisarmos bem, os propósitos de Aml em si apresentam problemas multiobjetivos como, por exemplo, o uso de tecnologia que responda a necessidades do usuário, porém de forma transparente. Sabe-se que 'uso de tecnologia' e 'transparência' podem ser ideias conflitantes, afinal como execu-

tar uma ação no ambiente sem que o usuário perceba a interferência da tecnologia neste? De certa forma, estes objetivos são também incomparáveis, não sendo conveniente a priorização de um ou outro aspecto. Considerando os aspectos relacionados à transparência, aplicações ubíquas e pervasivas (AUGUSTO; MCCULLAGH, 2007) visam oferecer uma solução em Aml. Muitas vezes, com o objetivo de tomar uma decisão para reagir a um fato ou antecipar uma ação do usuário, estas aplicações buscam imitar o raciocínio humano, o qual, por natureza, busca contemplar diversos objetivos para tomar a decisão mais conveniente para uma situação. Entretanto, este raciocínio não é exato, ele é vago, complexo e não linear, tornando-se difícil de ser reproduzido computacionalmente com a utilização de lógica clássica.

Além da reprodução do raciocínio humano, sistemas em Aml podem demandar o raciocínio sobre outros problemas complexos como, em um exemplo frequente, buscar o conforto do usuário enquanto busca-se a otimização de recursos. Estes aspectos complexos demandam o uso de modelos mais realistas e que reproduzam mais fielmente aspectos dinâmicos tornando, desta forma, o uso de modelos lineares simplista e inadequado para este contexto (SANTOS et al., 2007). Como consequência, enquanto está clara a necessidade de modelos que considerem estes aspectos no desenvolvimento de aplicações com alto grau de inteligência, ainda não é muito óbvio como desenvolver estes modelos. Outra questão a ser observada é que, devido a diversos fatores como complexidade, falta de tempo de processamento, falta de dados, imprecisão dos dados, etc., diversas decisões no mundo real são feitas com certo grau de incerteza ou imprecisão. Com isso, são necessários métodos específicos que considerem todos estes fatores facilitando o raciocínio computacional.

A lógica fuzzy (ZADEH, 1965) busca tornar sistemas capazes de raciocinarem de uma maneira aproximada ao raciocínio de um ser humano. Em outras palavras, a lógica fuzzy é um modelo matemático que visa possibilitar que um computador reproduza o raciocínio humano na modelagem do mundo real. A lógica fuzzy permite o raciocínio em contextos vagos e incertos, fazendo uso de conceitos vagos normalmente utilizados em nosso cotidiano como, por exemplo, 'alto', 'calor', 'frio'... Pelo uso de uma linguagem natural e do conceito de grau, este tipo de lógica visa atingir resultados muito mais próximos a realidade, se comparado ao que seria obtido com o uso da lógica clássica. Entretanto, a limitação da maioria dos controladores fuzzy é que, apesar de permitirem o raciocínio sobre diferentes valores de entrada, geralmente, isto é feito dando prioridade a um ou outro elemento.

Desta forma, o conceito fuzzy foi generalizado em 1967 para conjuntos \mathcal{L} -fuzzy para um *Brouwerian lattice* arbitrário completo (\mathcal{L}) ao invés do intervalo de unidade $[0, 1]$ dos números reais. Winter (WINTER, 2007) introduz a teoria e provas matemáticas necessárias para o raciocínio \mathcal{L} -fuzzy baseado na teoria das categorias Goguen. A teoria \mathcal{L} -fuzzy consiste na aplicação da teoria dos *lattices* na lógica fuzzy com o objetivo de lidar

com contextos multiobjetivos e atingir a solução mais adequada sem priorizar um ou outro aspecto. Com o uso de conceitos e propriedades de um *lattice*, é possível o raciocínio em contextos não lineares e a obtenção de resultados baseados em todos os critérios importantes necessários ao contexto.

Além do uso de metodologias como \mathcal{L} -fuzzy, o conhecimento do contexto permite que as aplicações se tornem mais flexíveis e adaptáveis às necessidades do usuário, resultando em um raciocínio mais adequado e agindo em benefício deste. Aml explora o uso de sensores capazes de captar dados sobre o contexto como, por exemplo, formas, movimento, temperatura, reconhecimento de som. Esses sensores podem produzir dados brutos e ao serem processados podem produzir informação relevante como, por exemplo, sensores de corrente e tensão utilizados para gerar o consumo de energia de um aparelho elétrico. Estes dados possibilitam a identificação de situações de interesse ao sistema como, por exemplo, possíveis riscos ou a identificação de que o ambiente não está adequado às preferências do usuário. Sistemas Sensíveis à Situação (MACHADO, 2015) devem concentrar seu foco em informações de alto nível conceitual ao invés de dados de baixo nível. No entanto, sensores são capazes de produzir apenas dados do contexto, e não informação. Ou seja, estes dados não apresentam semântica que os dê um significado. Uma maneira de prover semântica aos dados obtidos dos sensores é por meio do uso de modelos que identificam o contexto de interesse (MACHADO et al., 2017). Por meio da identificação de entidades relevantes e seus relacionamentos, o uso destes modelos permite que sejam obtidos apenas os dados relevantes ao domínio do sistema. Estes sistemas são conhecidos como sistemas sensíveis ao contexto (SCHILIT; ADAMS; WANT, 1994).

Considerando os aspectos citados, este trabalho apresenta um modelo sensível a situação para auxiliar na tomada de decisão multiobjetiva para objetivos com mesma importância em Aml. Este modelo pode ser utilizado no desenvolvimento de sistemas em Aml que buscam reproduzir o raciocínio humano na solução de problemas multiobjetivos. O modelo busca dar a estes sistemas a capacidade de identificar a ocorrência de uma situação de interesse e disparar a tomada de decisão multiobjetiva com base no contexto do domínio. Para tomada de decisão, o módulo de decisão consiste em um controlador \mathcal{L} -fuzzy, o qual processa dados relacionados aos múltiplos objetivos sem atribuir pesos aos objetivos. A partir da decisão tomada, uma ação pode ser executada com o objetivo de manipular a situação atual do ambiente.

Para definição do modelo proposto, este trabalho utiliza modelos e teorias definidos nos trabalhos de Machado (MACHADO et al., 2017) e Winter (WINTER, 2007) e para sua verificação, foi desenvolvido um sistema para o cenário de uma sala utilizada como escritório em um Aml, para a qual deseja-se contemplar os objetivos: conforto térmico e eficiência energética. Consideramos que o usuário, um programador, trabalha em casa gastando maior parte de seu tempo neste cômodo. Considerando que ele vive em um

local de clima quente e, com o objetivo de deixar o ambiente termicamente agradável, ele mantém o ar-condicionado constantemente ligado em 22° gerando, assim, um consumo alto de energia (comportamento padrão em países tropicais). Deseja-se adicionar inteligência ao ambiente reproduzindo decisões que seriam tomadas por um especialista ao tentar evitar um consumo energético excessivo ao mesmo tempo que objetiva-se controlar a temperatura mantendo o conforto térmico adequado.

Os resultados obtidos com a utilização da aplicação neste ambiente foram comparados com dois outros cenários de teste: a) um cenário para o qual não foi utilizada a aplicação e, portanto, não foi aplicada nenhuma ação para controle dos objetivos e b) um cenário utilizando o mesmo método de tomada de decisão, porém sem detectar a situação de interesse.

Além do modelo definido, este trabalho disponibiliza uma biblioteca para implementação de controladores \mathcal{L} -Fuzzy (WINTER, 2007), a qual foi utilizada para tomada de decisão multiobjetiva. Para coleta dos dados relacionados ao consumo energético e à temperatura do ambiente, este trabalho utilizou um dispositivo IoT, desenvolvido em outro projeto de pesquisa. Além da coleta destes dados, este dispositivo é capaz manipular a configuração de temperatura do ar condicionado conforme definido pelo sistema desenvolvido.

1.1 OBJETIVOS E CONTRIBUIÇÕES

Este trabalho aborda a tomada de decisão multiobjetiva em Ambientes Inteligentes (Aml) visando, sempre que detectada uma situação de interesse, identificar uma solução adequada para manipular essa situação. O trabalho tem como objetivo propor uma abordagem para sistemas agirem frente a essas situações, buscando soluções que contemplem multiobjetividade. Portanto, a questão que motiva a presente pesquisa é: uma abordagem sensível a situação suportada pela teoria das relações \mathcal{L} -fuzzy para tomada de decisão multiobjetiva permite a manipulação de situações indesejadas em Ambientes Inteligentes?

Neste trabalho, o **objetivo** é desenvolver um modelo que leve em consideração o contexto do usuário e possa ser usado em sistemas para ambientes inteligentes que visam controlar situações relacionadas a problemas de caráter multiobjetivo.

Os objetivos específicos referem-se:

- formalização do modelo proposto em Machado (MACHADO et al., 2017) para identificação da situação de interesse, a qual evidencia a necessidade da tomada de decisão em um contexto específico;
- desenvolvimento de uma biblioteca genérica para controladores \mathcal{L} -fuzzy, tendo como características essenciais a multiobjetividade, possibilidade de uso de intervalos não

lineares e totalidade, ou seja, para cada valor de entrada haverá um valor de saída. Controladores \mathcal{L} -fuzzy correspondem ao módulo de decisão;

- implementação de protótipo do controlador \mathcal{L} -fuzzy utilizando a biblioteca desenvolvida para validação e verificação de generalidade do modelo.

As principais contribuições deste trabalho são: a disponibilização do modelo desenvolvido, o qual foi verificado apresentando resultados satisfatórios; a disponibilização da biblioteca \mathcal{L} -fuzzy, que pode ser utilizada para desenvolvimento de controladores para tomada de decisão multiobjetiva.

1.2 ORGANIZAÇÃO DO TEXTO

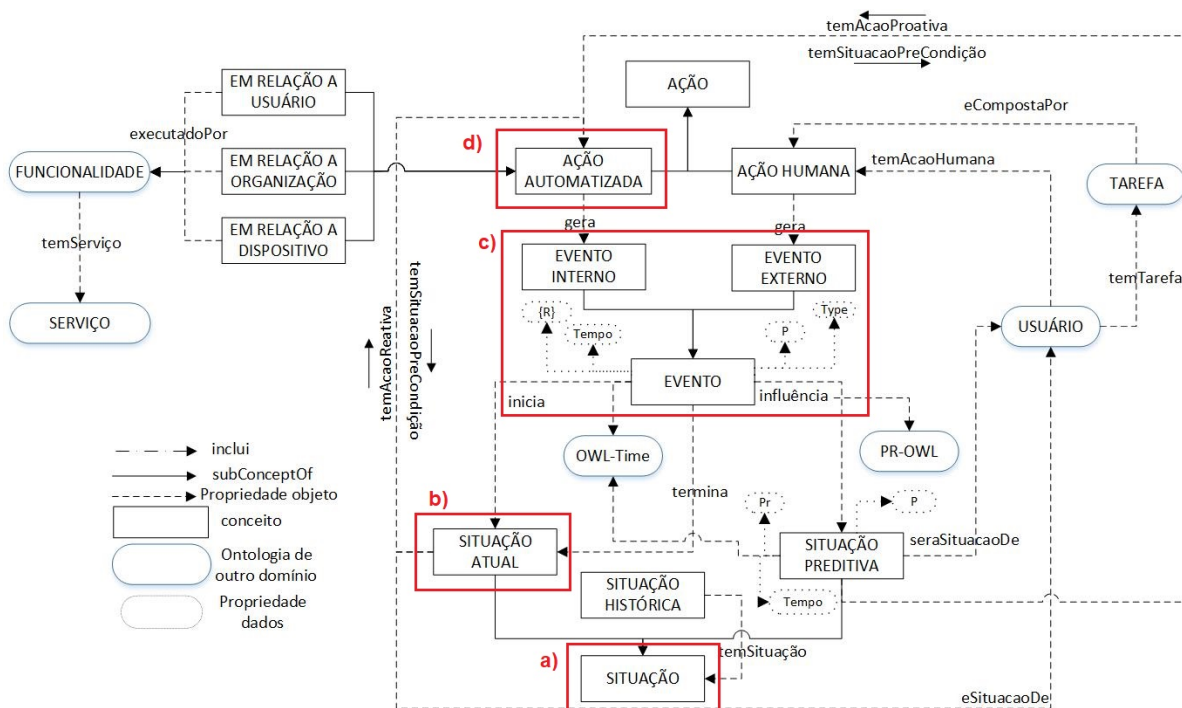
O texto desta dissertação está organizado da seguinte forma:

O Capítulo 1 introduz o estudo realizado neste trabalho. O Capítulo 2, intitulado Referencial Teórico, descreve os principais fundamentos teóricos envolvidos na definição do modelo. O Capítulo 3 apresenta trabalhos relacionados aos diferentes conceitos e tecnologias utilizados neste trabalho. O Capítulo 4, denominado Modelo Proposto, visa descrever os detalhes do modelo desenvolvido neste trabalho. Um estudo de caso envolvendo o problema multiobjetivo conforto térmico e eficiência energética é descrito no Capítulo 5. Por fim, os resultados do estudo de caso são apresentados no Capítulo 6 e o Capítulo 7 conclui o trabalho.

2 REFERENCIAL TEÓRICO

Esta seção apresenta o referencial teórico necessário para compreensão do modelo definido neste trabalho. Além de conhecer algumas metodologias para tomada de decisão multiobjetiva, é necessária a compreensão de alguns conceitos como o conceito de contexto, situação, evento e ação, além da compreensão da relação existente entre eles. Machado (MACHADO et al., 2017) demonstra esta relação por meio da definição de um modelo conceitual (Figura 2.1) que baseia-se na definição destes conceitos para modelagem de um domínio proativo em Aml. O modelo desenvolvido neste trabalho complementa este modelo conceitual com aspectos necessários a tomada de decisão multiobjetiva em Aml. Para maior compreensão da modelo conceitual citado, ver (MACHADO, 2015).

Figura 2.1 – Modelo conceitual do domínio proativo.



Fonte: Adaptado de (MACHADO, 2015).

2.1 CONSCIÊNCIA SITUACIONAL

O termo 'consciência situacional' (*situation awareness - SA*) tem origem na aviação, a qual demanda a operação e controle de sistemas complexos em um ambiente dinâmico (ENDSLEY; GARLAND, 2000). Além da aviação, SA tem se estendido para outros domínios que, dentre outros aspectos, apresentam algumas características específicas como:

ambientes dinâmicos com alto grau de informação; domínios onde forte treinamento é demandado; o tempo disponível para ação é restrito (UHLARIK; COMERFORD, 2002). Neste contexto, algumas das áreas que fazem grande uso e estudam o domínio de SA são a educação, condução de veículos, operações em usinas de energia, previsão meteorológica e até mesmo áreas como autodefesa e equipes esportivas profissionais ou recreativas (ENDSLEY, 2011). Como consequência da necessidade de novas ferramentas e *frameworks* que atendam a crescente demanda de sistemas eletrônicos e de automação que ofereçam suporte aos domínios citados, SA tem ganho forte ênfase na tecnologia. No entanto, deve-se observar que sistemas tecnológicos não geram SA (ENDSLEY; GARLAND, 2000), o papel da tecnologia em SA consiste em facilitar a detecção de situações por meio de sistemas de suporte a decisão.

Existe ainda certa discordância quanto à definição exata de SA. Uhlarik e Comerford (UHLARIK; COMERFORD, 2002) descrevem SA como algo gerado psicologicamente e que, conseqüentemente, não pode ser claramente perceptível dificultando sua definição. Endsley (ENDSLEY, 2011) se refere à SA como o ato de “*estar atento ao que está acontecendo a sua volta e compreender o que esta informação significa para você agora e no futuro*”. De forma similar e mesmo ainda não sendo aceita (UHLARIK; COMERFORD, 2002) por muitos autores, a definição mais citada na literatura foi proposta pela pesquisadora Mica R. Endsley e diz que SA consiste na “*percepção dos elementos em um ambiente dentro de um volume de tempo e espaço, a compreensão de seu significado e a projeção de seu estado em um futuro próximo*” (ENDSLEY, 1995).

Para melhor compreensão do significado de SA podem ser revistos os significados de ambos termos separadamente. Situação é descrita por Machado (MACHADO et al., 2017) como um conjunto de entidades ativas e as interações entre estas em um intervalo de tempo. Em outras palavras, uma situação tem um tempo de início e fim, é composta de entidades, seus atributos e as relações entre estas entidades. Situação corresponde a uma parte limitada (se detém aos aspectos relevantes e que contém algum significado) de um todo e pode ser identificada na camada de Compreensão descrita por Endsley (descrita mais adiante). Aspectos que geram uma situação não se limitam a um espaço geográfico podendo estes ser físicos, conceituais ou ambos (KOKAR; MATHEUS; BACLAWSKI, 2009). *Awareness*, por sua vez, pode ser traduzido como ‘*estado de ser cômico, ciente, atento*’¹ e, como descrito em (KOKAR; MATHEUS; BACLAWSKI, 2009), implica mais do que apenas observar algum objeto ou elemento, implica na geração de inferências a partir do que foi observado.

O conceito de SA é normalmente aplicado em situações específicas, onde se aplica a informação do contexto (definido por (ABOWD et al., 1999) como: “*qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade*”) para alcançar um objetivo claro ou realizar uma tarefa específica. A informação aplicada consiste apenas na

¹<http://www.michaelis.com.br>

parte da informação do contexto que é relevante para a tarefa, ou seja, ignora-se qualquer informação que não apresente relevância direta para a situação. Sistemas atuais são capazes de gerar um vasto número de dados. Entretanto, como apontado em (ENDSLEY; GARLAND, 2000), 'mais dados' é algo diferente de 'mais informação', ou seja, uma alta quantidade de dados não implica necessariamente em mais informação. Se estes dados não forem apresentados da forma correta, eles serão, muitas vezes, inúteis. Em síntese, o problema de sistemas atuais não é a falta de informação e sim encontrar a informação necessária quando ela é necessária.

No modelo de Endsley, SA compreende três níveis: (a) percepção dos elementos no ambiente, (b) compreensão de situação atual e (c) projeção do estado futuro. Estes níveis são descritos a seguir.

1. Percepção dos elementos no ambiente

O primeiro passo em SA compreende a detecção do *status*, atributos e outros aspectos relevantes dos elementos no ambiente. A percepção destes atributos ocorre de forma diferente para cada domínio em questão, ou seja, elementos considerados importantes para um dado domínio, podem não apresentar relevância alguma para outro domínio do mesmo contexto. Um ciclista, por exemplo, mesmo estando em um ambiente similar ao de um pedestre, precisa estar atento a elementos diferentes dos observados por este pedestre (velocidade, obstáculos à frente, demais ciclistas, etc.).

A percepção da informação pode ocorrer de forma visual, verbal, audível, tato, paladar, olfato ou uma combinação destas (ENDSLEY, 2011). A exatidão e consequente confiabilidade desta informação é um fator crítico nesta etapa. Muitas vezes, a percepção de todos os elementos do ambiente que são relevantes ao domínio é altamente desafiadora, seja pela falta de conhecimento de sua importância ou por alguma limitação do observador. A percepção incorreta da informação aumenta drasticamente a possibilidade de uma identificação incorreta da situação (ENDSLEY; GARLAND, 2000).

Endsley (ENDSLEY, 2011) cita o exemplo da aviação onde a maioria dos problemas com SA ocorre nesta etapa. Em alguns casos, foi constatado que o problema ocorreu porque a pessoa responsável não recebeu a informação necessária ou devido a limitações do sistema utilizado. Em outros casos, a informação chave se encontrava disponível, porém não foi percebida por conta de distrações.

Um sistema SA deve garantir que a informação necessária seja obtida e processada corretamente, seja por um usuário ou pelo próprio sistema.

2. Compreensão de situação atual

A segunda etapa de SA consiste no entendimento da informação obtida na primeira

etapa (ENDSLEY, 2011). Em outras palavras, nesta etapa os dados obtidos anteriormente são juntados e processados com o objetivo de compreender sua importância/relevância e significado para o objetivo/tarefa em questão. Segundo Endsley (ENDSLEY, 2011), a compreensão destes dados é trabalhosa e exige uma boa base de conhecimento. Alguém com pouco conhecimento ou experiência no domínio certamente apresentará maior dificuldade nesta compreensão do que alguém com maior compreensão do domínio podendo, até mesmo, resultar em inconsistências.

3. Projeção do estado futuro

De acordo com Endsley (ENDSLEY, 2011), só é possível chegar a esta etapa com uma boa compreensão do domínio e situação em questão. Na projeção, são juntadas as informações atuais que o operador tem sobre o domínio com os modelos mentais (descritos mais adiante) existentes sobre este com o propósito de diagnosticar o que acontecerá a seguir. Esta etapa ocorre após a compreensão dos elementos do ambiente e como eles afetam o objetivo atual. Com isto, é possível prever como eles afetarão o ambiente/domínio no futuro próximo. Baseada em sua experiência, Endsley compactua com trabalhos que afirmam que operadores experientes se baseiam fortemente em projeções futuras (ENDSLEY; GARLAND, 2000).

Geralmente, falhas na etapa de projeção ocorrem devido à falta de conhecimento sobre o domínio ou a falta de recursos sobre este (ENDSLEY, 2011). Além disto, (PANTELI et al., 2013) cita aplicações de software (aplicações utilizadas pelos operadores de sistema para monitorar e controlar o sistema), medidas em tempo real (dados inconsistentes, faltantes ou conflitantes), fatores do ambiente (fatores que geram confusão e induzem a negligência de informação crítica, automação (possíveis falhas em sistemas automatizados podem passar despercebidas), fatores individuais (experiência, treinamento e alerta do operador) e, finalmente, problemas de comunicação com outros operadores como fatores que influenciam na insuficiência de SA.

Para exemplificar os três níveis descritos, imagine o cenário do ciclista que se depara com um obstáculo à sua frente. Se o ciclista vê o obstáculo, então ele percebeu o elemento na situação atual (nível 1), ao reconhecer que este obstáculo representa um perigo, ele compreendeu a situação (nível 2), ao estimar o tempo restante que ele tem antes de colidir com o obstáculo e realizar uma manobra de desvio, ele projetou o estado futuro (nível 3).

Tempo exerce um papel importante em SA (ENDSLEY, 2011), especialmente nos níveis 2 e 3. Sua compreensão é importante devido ao fato de que os elementos do ambiente mudam seu estado constantemente e o sistema deve se adaptar a este novo estado. Um aspecto crítico relacionado ao tempo em SA é a identificação de quanto tempo se tem disponível antes que determinado evento ocorra e alguma ação deva ser executada. Como aspectos do mundo real sofrem constante mudança, com o objetivo de preservar SA,

um operador humano ou até mesmo um sistema são forçados a atualizar suas estratégias de percepção da informação constantemente.

Outro aspecto importante em SA é percepção. Endsley (ENDSLEY, 2011) relata que a capacidade de percepção humana é limitada. Um indivíduo não possui a habilidade de perceber todos os elementos importantes a sua volta simultaneamente. Ainda segundo a autora, um indivíduo desenvolve mecanismos conhecidos como modelos mentais, os quais correspondem a um entendimento sistemático de como algo funciona e são incrementados com base em experiências vividas por este indivíduo. Modelos mentais tornam possível que uma pessoa identifique quais elementos são importantes e compreendam sua importância ajudando-a a formar um raciocínio. Neste contexto, SA pode ser entendido como um modelo mental internalizado do estado atual do indivíduo (ENDSLEY; JONES, 2013).

Um operador de um sistema complexo recebe dados de diversas fontes: do ambiente a sua volta, sistemas, outras pessoas, etc. Entretanto, como mencionado anteriormente, informação é percebida por ele de acordo com o objetivo que ele está direcionado e suas expectativas. Para evitar falhas significativas em SA, este operador deve ser suficientemente responsivo ao ponto de perceber quando uma situação mais importante surge e seu objetivo deve ser mudado, bem como, a informação que ele precisa perceber no ambiente. O entendimento de como uma pessoa distingue e utiliza objetivos se faz importante para o *designer* de um sistema, pois com esta compreensão é possível saber quais informações são importantes para o operador em determinada situação. Sem a compreensão do impacto da situação nos objetivos do operador em SA, é possível que um sistema apresente informações sobre o domínio que não contenham significado algum, mesmo que estas sejam informações corretas. Neste contexto, surgem os sistemas sensíveis ao contexto e à situação.

2.1.1 Sensibilidade ao contexto e situação

Em sistemas inteligentes, entidades podem ser identificadas como, por exemplo, pessoas, sensores, lugares ou objetos relevantes para a interação entre o usuário e a aplicação. Os autores em Abowd et al. (1999) caracterizam o próprio usuário e a aplicação como entidades. Entidades não necessitam ser necessariamente físicas, elas podem ser elementos abstratos (hora, data, localização do usuário) desde que apresentem relevância ou forneçam informação relevante ao contexto. Conhecer as entidades e as relações existentes entre estas no ambiente para o qual a aplicação é destinada, faz com que as aplicações possam agir e interagir de forma mais vantajosa em relação a seu domínio. Sistemas que fazem uso do contexto visando atingir estes objetivos são conhecidos como sistemas sensíveis ao contexto (SCHILIT; ADAMS; WANT, 1994).

Machado (MACHADO et al., 2017) define um modelo para a representação computacional do contexto (2.1). Segundo o autor, o contexto do ambiente é caracterizado por entidades e suas relações semânticas. Neste modelo, as relações semânticas (R) que formam o contexto são representadas como triplas:

$$\langle Es, p, Eo \rangle \quad (2.1)$$

onde Es corresponde a entidade que age como sujeito, Eo a entidade de age como objeto e p representa um predicado de contexto que encapsula duas entidades de contexto em uma relação. Por exemplo, a relação semântica: $\langle \text{objeto}, \text{temSensor}, \text{sensor} \rangle$ apresenta entidades genéricas ligadas ao predicado temSensor . De acordo com o contexto do ambiente para qual a aplicação é desenvolvida, estas entidades (objeto e sensor) devem ser instanciadas como, por exemplo: $\langle \text{arCondicionadoSala}, \text{temSensor}, \text{DHT11} \rangle$.

No entanto, deve ser observado que um sistema não é capaz (ou não necessita) identificar e processar todas as entidades de um ambiente. Consequentemente, faz-se necessário realizar a modelagem do contexto identificando apenas o conjunto das entidades físicas ou abstratas que apresentam relevância ao problema definido. Esta parte do contexto que interessa a aplicação sendo desenvolvida é chamada *contexto de interesse* (C). Diferentes aplicações apresentarão contextos de interesse diferentes, de acordo com seu domínio. Machado (MACHADO et al., 2017) representa o contexto de interesse da seguinte forma (2.2):

$$C : (app, \{R\}) \quad (2.2)$$

onde C representa o contexto de interesse, app representa uma aplicação específica e $\{R\}$ consiste em um grupo de relações semânticas relevantes a este contexto. Este contexto é a informação de interesse utilizada pela aplicação para a realização do processamento necessário para atingir seu objetivo.

Aplicações utilizadas em ambientes inteligentes, normalmente fazem uso de sensores que coletam dados sobre as entidades do ambiente. Entretanto, estes dados são obtidos em baixo nível, não oferecendo semântica ou significado claro à aplicação. Desta forma, com a informação gerada sobre o contexto, os dados podem ser obtidos e agregados gerando informação útil, a qual pode ser utilizada para detectar a situação atual do ambiente (informação de mais alto nível).

De acordo com (MACHADO, 2015) Sistemas Sensíveis à Situação para Ambientes Inteligentes devem concentrar seu foco em informações de alto nível conceitual ao invés de dados de baixo nível, como dados de sensores. Por exemplo, um sistema que monitora um paciente deve interpretar se este está sofrendo um ataque cardíaco ao invés de apenas transmitir dados como valores de seus batimentos cardíacos. Portanto, para atingir este objetivo, é necessário que estes dados sejam interpretados de uma forma que identifique

a situação atual do ambiente. Como pode ser visualizado na Figura 2.1, Machado (MACHADO et al., 2017) determina que o início e o fim da situação atual são determinados por eventos (*b* e *c*).

Evento (*c*) pode ser definido como ‘algo’ que ocorre em um instante no tempo. Ou seja, um acontecimento registrado em um tempo específico, ao invés da utilização de registros de início e fim. A ativação de uma funcionalidade, o envio de uma mensagem e a emissão de um sinal sonoro são exemplos de eventos comuns em sistemas Aml relacionados a saúde. Estes eventos podem ser internos ou externos (MACHADO et al., 2017), sendo que eventos internos são aqueles produzidos pelo sistema, e externos são os gerados pela interação de um usuário com a aplicação. De acordo com (MACHADO et al., 2017), eventos podem ser representados conforme a declaração (2.3).

$$\textit{Evento} : (\textit{nome}, \textit{tipo}, \textit{tempo}, \{R\}, p) \quad (2.3)$$

onde o evento é composto por um nome, tipo (interno ou externo) (valor opcional), o registro do tempo em que ele ocorreu, um conjunto de relações semânticas que o descrevem, e um padrão *p*, o qual é utilizado para detecção do evento quando este não pode ser detectado pela simples coleta de um dado ou consumo de um serviço.

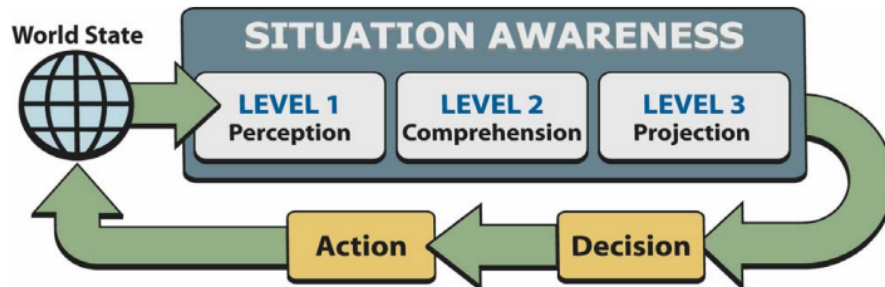
Logo, a detecção da situação atual (*Sa*) (Figura 2.1-*b*) no ambiente está diretamente ligada a ocorrência de dois eventos: evento de início (*Ie*) e evento de fim (*Fe*). Neste trabalho, consideramos o modelo de *Situação Atual* utilizado em (MACHADO et al., 2017), o qual é baseado no trabalho de (PERNAS et al., 2014). Este modelo segue a seguinte definição (2.4):

$$\textit{Sa} : (\textit{nome}, \textit{Ie}, \textit{Fe}) \quad (2.4)$$

Sensibilidade a situação é considerado o principal pioneiro em tomada de decisão (ENDSLEY, 1995) e, por compreender a mudança do conhecimento sobre o contexto, SA é essencial para tomada de decisão efetiva. Contudo, como pode ser observado na Figura 2.2, SA é considerada apenas uma etapa na tomada da tomada de decisão. Uma das razões é o fato de ser possível que a situação seja identificada corretamente e ainda assim seja tomada a decisão incorreta. Isto pode ocorrer devido a diversos fatores, como falta de experiência, fatores individuais de personalidade, limitações técnicas ou organizacionais, falta de treinamento, etc.. Endsley (ENDSLEY; GARLAND, 2000) relata concordar com pesquisas que afirmam que decisões são formadas por SA e SA é formado por decisões. Entretanto, apesar desta ligação, a autora ressalta que tomada de decisão e SA não precisam estar atrelados como um único processo e é muito comum que, na prática, isto realmente não ocorra.

Desta forma, aliado a SA torna-se adequado o uso de mecanismos que visam reproduzir a forma humana de raciocinar. Além disso, Machado (MACHADO, 2015) destaca que a detecção de situações apresenta aspectos incertos. Isto ocorre tanto devido à vul-

Figura 2.2 – SA e tomada de decisão.



Fonte: (PANTELI et al., 2013).

nerabilidade dos dados obtidos pelos sensores quanto a possibilidade de erros na própria modelagem humana. Desta forma, o raciocínio para a tomada de decisão neste contexto deve ser realizado por meio do uso de tecnologias que levem em consideração o aspecto incerteza. Como o foco deste trabalho está em problemas multiobjetivos, a próxima seção apresenta algumas metodologias utilizadas na tomada de decisão multiobjetiva em Ambientes Inteligentes, incluindo alternativas que incluem a incerteza no raciocínio.

2.2 TOMADA DE DECISÃO MULTICRITÉRIOS

Existem casos de tomada de decisão que apresentam mais de um objetivo a ser contemplado, sendo necessário trabalhar com parâmetros incomparáveis a fim de obter um único resultado em um contexto. Por exemplo, supondo que se busque adquirir um eletrodoméstico que possua ambas qualidades: baixo consumo energético e alta eficiência, uma opção para esta escolha é analisar ambas características separadamente e escolher a de maior valor, por exemplo. Entretanto, por que baixo consumo deveria ter prioridade sobre o outro aspecto ou vice-versa? Quando ambos aspectos são relevantes, um não pode (e não deve) ser priorizado. Neste contexto, surge o conceito de 'multiobjetividade'.

Quando falamos em escolhas e problemas do mundo real, é comum que estes envolvam múltiplos objetivos e que, muitas vezes, estes objetivos sejam conflitantes. Isto ocorre nas mais diversas áreas como eficiência energética, Aml, engenharia, economia e até nas questões mais simples do dia a dia. Ao ir ao supermercado, por exemplo, é comum buscar produtos de qualidade com o melhor preço. Ao dirigir, normalmente, buscamos chegar ao destino rapidamente, mas sem deixar de lado o quesito segurança. O raciocínio humano busca contemplar estes diversos objetivos para tomar a decisão mais conveniente a cada situação. Entretanto, este raciocínio não é exato, ele é vago e complexo tornando-

se difícil de ser reproduzido computacionalmente com a utilização de lógica clássica.

A Tomada de Decisão com Múltiplos Critérios (*Multiple Criteria Decision Making*, MCDM) (MIETTINEN, 2012) é uma metodologia utilizada com problemas multiobjetivos e multicritérios. MCDM é uma área que ganhou foco no fim do século 19 nos trabalhos sobre economia de Edgeworth e Pareto (EL-WAHED, 2008) e apresentou forte expansão nos anos 90 (MIETTINEN, 2012). Apesar de ser fortemente aplicada na solução de problemas complexos em engenharia, (EL-WAHED, 2008) destaca MCDM como sendo também adequada para lidar com os mais diversos problemas da vida real, pois estes geralmente apresentam objetivos conflitantes. MCDM é dividida em dois subtipos:

- *Multiattribute Decision Analysis* ou *Multiple Attribute Decision Making* (MADMA): o conjunto de alternativas viáveis é discreto, predeterminado e finito (MIETTINEN, 2012). MADM faz seleções entre alguns cursos de ação na presença de muitos e, geralmente, conflitantes atributos (critérios). Assume-se que problemas MADM tenham um número de alternativas de decisão limitado e predeterminado. Metodologias MADM podem ser vistas como métodos alternativos que combinam a informação em uma matriz de decisão junto a informação obtida com o tomador de decisão (uma pessoa, neste caso) com o objetivo de identificar o resultado final (KAHRAMAN, 2008). Normalmente, metodologias MADM apresentam a utilização de pesos de importância para os atributos. Um exemplo de MADM é a escolha do eletrodoméstico que melhor satisfaça os atributos citados anteriormente, dado um conjunto contendo algumas opções de eletrodomésticos, ou seja, para a tomada de decisão existe um conjunto finito com opções conhecidas.
- *Multiple Objective Decision Making* (MODM): as alternativas viáveis não são explicitamente conhecidas antecipadamente. Existe um número infinito de alternativas e estas são representadas por variáveis de decisão que apresentam funções de restrição (MIETTINEN, 2012). MODM utiliza um *framework* matemático para identificação das alternativas e cada alternativa é avaliada conforme quanto ela satisfaz um ou múltiplos objetivos (critérios) (KAHRAMAN, 2008). Normalmente, metodologias MODM buscam maximizar ou minimizar os objetivos propostos (maximizar eficiência enquanto o consumo energético é minimizado), não considerando um objetivo mais importante que o outro. O uso de objetivos não deixa claro, porém torna implícitas as variáveis que serão utilizadas como entrada, por exemplo, qual variável/variáveis devem ser utilizadas para que o objetivo eficiência seja contemplado? Seguindo o exemplo anterior, uma tomada de decisão multiobjetiva ocorreria quando, primeiramente você decide os objetivos que o eletrodoméstico deve cumprir. Com isso, você calcula as configurações necessárias para atingir estes objetivos e em seguida você vai a uma loja e pergunta quais eletrodomésticos eles têm que correspondam a estas configurações. Por fim, qualquer um destes eletrodomésticos será adequado.

Por necessitar de um maior conhecimento sobre os objetivos, MODM exige uma interação muito maior com o tomador de decisão (especialista) do que MADM. Em resumo, MADM tem como entrada um conjunto predeterminado de alternativas, para as quais será tomada uma decisão escolhendo uma única alternativa como valor de saída. MODM, por sua vez, tem como entrada os objetivos que devem ser considerados e apresenta como saída um conjunto com todas alternativas consideradas igualmente adequadas para atender estes objetivos, sendo que qualquer alternativa pode ser selecionada como resultado final. Neste trabalho foi abordada a tomada de decisão multiobjetiva (MODM).

Em geral, problemas de otimização multiobjetiva são solucionados por escalabilidade (MIETTINEN, 2012), ou seja, o problema é conferido em um único ou em uma família de problemas de otimização com objetivo único. Após esta conversão, podem ser aplicados métodos de otimização para problemas com objetivo único. Entretanto, devido a aspectos como diferenças de unidades, características conflitantes, etc., nem sempre é possível encontrar uma única solução adequada com relação a todos objetivos apresentados, ao contrário, existe um conjunto de soluções aceitáveis. Este conjunto é conhecido como o conjunto *Pareto* (NGATCHOU; ZAREI; EL-SHARKAWI, 2005) e, para este, busca-se melhorar (maximizar ou minimizar) um objetivo sem prejudicar outro sendo que todas as soluções são consideradas igualmente boas. No entanto, nem sempre deseja-se maximizar ou minimizar um objetivo e, além do problema da dificuldade em encontrar uma única solução, muitos problemas multiobjetivos não são lineares. Portanto, são necessários novos conceitos, métodos e maneiras de pensar que os manipulem eficientemente, pois métodos existentes para otimização de objetivo único não são suficientes.

Diversas abordagens podem ser encontradas na literatura para manipulação de problemas multiobjetivos. A maioria dos trabalhos nesta área fazem uso de algoritmos genéticos e otimização pareto como, por exemplo, os algoritmos genéticos EMO (*evolutionary multi-objective optimization*) (DEB, 2011), que contemplam um número de funções que necessitam ser minimizadas ou maximizadas. O algoritmo pode conter um número de restrições das quais qualquer solução possível deve satisfazer. As soluções ótimas podem ser definidas a partir do conceito matemático de ordenação parcial. Um conjunto solução é definido a partir de dois aspectos: (a) a solução x não é inferior a solução y em nenhum dos objetivos e (b) a solução x é estritamente melhor que a solução y em pelo menos um dos objetivos. Após a identificação do conjunto de soluções *pareto-optimal*, este necessita ser processado com o objetivo de identificar um único resultado adequado. Para alcançar este objetivo, é comum o uso de algoritmos como PSO (WANG; YANG; WANG, 2010), MOGA (SHAIKH et al., 2016) ou TOPSIS (GOVINDAN; KHODAVERDI; JAFARIAN, 2013). Uma limitação em EMO é o fato de que sua eficiência diminui conforme o número de objetivos aumenta. Isto ocorre pois, as populações passam a não dominar uma a outra (DEB, 2011).

Na tomada de decisão no mundo real, diversas vezes os valores de entrada, ou até

mesmo valores de saída esperados são vagos, resultando na necessidade de uma metodologia que se adapte a estas situações. Métodos MCDM clássicos não são apropriados para processamento de dados incertos ou vagos. Além disso, os autores em (GOVINDAN; KHODAVERDI; JAFARIAN, 2013) ressaltam que métodos de tomada de decisão multicritérios no mundo real frequentemente se deparam com preferências humanas subjetivas. Como estas preferências são vagas e complexas, sendo difíceis de serem estimadas com exatidão, o uso de definições linguísticas se torna mais conveniente que o uso de valores exatos.

A lógica fuzzy é uma ferramenta adequada para este tipo de contexto (KAHRAMAN, 2008), surgindo, assim, o conceito de Tomada de Decisão Multicritérios *Fuzzy* (*Fuzzy Multi-criteria Decision Making*, FMCDM) (EL-WAHED, 2008), introduzidos em MCDM por (BELLMAN; ZADEH, 1970). A próxima seção descreve a lógica fuzzy e uma extensão desta, utilizada neste trabalho.

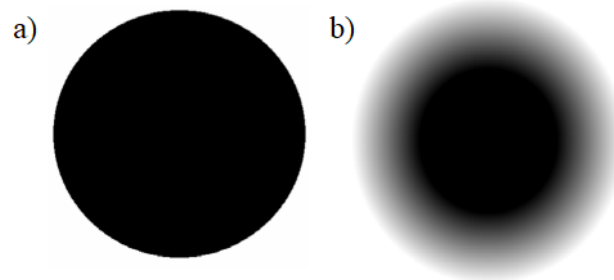
2.3 LÓGICA FUZZY

A lógica fuzzy é uma extensão da lógica clássica ou booleana, proposta por L.A. Zade em 1965 (ZADEH, 1965), a qual é baseada na teoria matemática de conjuntos fuzzy. Fuzzy é capaz de trabalhar com dados/situações incertos ou imprecisos, pois provê alta flexibilidade de raciocínio devido ao fato de trabalhar com o conceito de *grau de pertinência* (μ) (intervalo de unidade [0...1]) ao invés de se limitar a verdadeiro ou falso (1 ou 0). Em outras palavras, na lógica fuzzy tudo pode ser 'graduado', ou seja, fuzzy permite que seja usada uma palavra ou valor numérico indicando o grau de verdade da informação. Por exemplo, 25° apresenta um grau de pertinência de 0.7 no conjunto 'calor' ($\mu_{\text{calor}}=0.7$) e 0.1 no conjunto frio ($\mu_{\text{frio}}=0.1$). Desta forma, o elemento não precisa apenas pertencer (1) ou não pertencer (0) ao conjunto.

A lógica fuzzy utiliza uma linguagem natural, ou seja, humanamente compreensível. Os autores em (YAGER; ZADEH, 2012) destacam que em linguagem natural, a maioria dos predicados são apresentados em uma forma fuzzy (alto, doente, breve, rápido, muito maior que...). Desta forma, conceitos fuzzy são utilizados em nossa rotina diária. Por exemplo, se falarmos que estamos satisfeitos ou insatisfeitos com algo, que estamos cansados ou então que chegamos relativamente cedo ao trabalho, estaremos usando uma terminologia fuzzy. Os autores em (BAI; WANG, 2006) ressaltam que termos como estes podem ser criados e implementados apenas por seres humanos, o que quer dizer, computadores são incapazes de criar tais termos, pois compreendem apenas o significado de 0 e 1. Ou seja, fuzzy busca formalizar o raciocínio humano de uma forma mais natural do que simplesmente aplicando valores 0 ou 1, conhecidos por máquinas. Estes valores booleanos (0,1) são também chamados valores '*crisp*'. A figura 2.3 ilustra a diferença entre conjuntos *crisp* e conjuntos

fuzzy.

Figura 2.3 – Representação gráfica de um conjunto *crisp* (a) X um conjunto fuzzy (b).



Fonte: (DERNONCOURT, 2013).

Como visualizado na Figura, um conjunto *crisp* (a) apresenta apenas valores 0 e 1 para o valor 'preto'. O conjunto fuzzy (b), por sua vez, apresenta valores em um certo grau, gerando um aspecto sombreado no círculo. Observe que no centro do círculo o valor será 1 e na parte externa este valor é 0.

2.3.1 Etapas do processamento

Um processo fuzzy é um processo *crisp-fuzzy-crisp* (BAI; WANG, 2006), isto é, (i) recebe-se valores de entrada *crisp*, (ii) para serem processados, estes valores são convertidos para fuzzy, (iii) o resultado é convertido para *crisp* novamente. Isto ocorre pois, segundo (BAI; WANG, 2006), do ponto de vista do controle fuzzy ou da intuição de um ser humano, não existem valores absolutamente *crisp* no mundo real. O método mais conhecido para realização do processo de inferência fuzzy é o método *Mamdani* (MARRO et al., 2010).

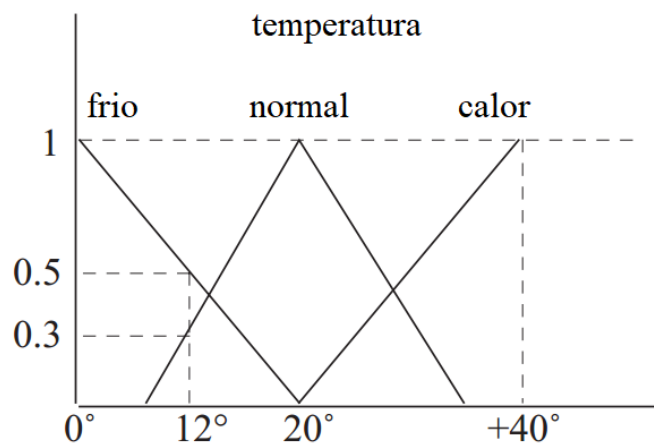
De acordo com o método *Mamdani*, as três etapas (BAI; WANG, 2006) necessárias para a implementação de Lógica fuzzy em uma aplicação são conhecidas como: (i) fuzzificação (conversão dos dados *crisp* em dados fuzzy; definição do grau de pertinência), (ii) processo de inferência fuzzy (combinar valores de pertinência obtidos na etapa anterior com as regras de inferência definidas a fim de obter uma saída fuzzy), (iii) defuzzificação (conversão do resultado fuzzy inferido para *crisp* novamente). Estas etapas são melhor detalhadas na sequência.

2.3.1.1 Fuzzificação

Com o objetivo de possibilitar que máquinas processem informações vagas como 'não muito alto' é necessário que se converta os dados de entrada e saída até então

numéricos (*crisp*) para variáveis linguísticas com componentes fuzzy. Para tal, no processo de fuzzificação primeiramente são definidas as funções de pertinência para cada possível variável de entrada e saída. Funções de pertinência (KLIR; YUAN, 1995) indicam o grau que um elemento pertence a um determinado conjunto. Por exemplo, para uma variável de entrada '*temperatura*' são definidos três variáveis linguísticas: frio, normal e calor. A Figura 2.4 ilustra o gráfico que representa a variável temperatura, identificando as variáveis linguísticas e as funções de pertinência para cada uma delas.

Figura 2.4 – Funções de pertinência para variável temperatura.



Fonte: Adaptado de (ROJAS, 2013).

De acordo com a figura acima, se for dado 12° como valor de entrada para temperatura, será identificado que a temperatura apresenta grau de pertinência (μ) de 0.5 para '*frio*', 0.3 para '*agradável*' e 0 para '*calor*'. De forma similar, se o valor de entrada for 40° os graus apresentados serão de 0, 0, 1, respectivamente.

A partir da fuzzificação, os valores tornam-se compatíveis para serem usados com as regras de inferência definidas para o domínio.

2.3.1.2 Processo de inferência fuzzy

Nesta etapa, com o objetivo de obter o conjunto de saída fuzzy, as regras de inferência são aplicadas aos valores linguísticos de entrada e saída. Regras de inferência (BAI; WANG, 2006) são regras baseadas no conhecimento ou experiência humanos sobre o domínio da aplicação. Elas são conhecidas como regras *if-then*, representando que ação deve ser tomada ou que informação deve ser obtida com base no valor de entrada. A

definição (2.5) representa estas regras:

$$IF\ x\ is\ M,\ THEN\ y = N \quad (2.5)$$

sendo x e y variáveis linguísticas de entrada e saída respectivamente, M e N entidades linguísticas correspondentes a conjuntos fuzzy. Estas regras devem ser definidas de acordo com a experiência e conhecimento do especialista e devem ser altamente intuitivas (BAI; WANG, 2006; VIEIRA et al., 1999) apresentando um aspecto facilmente entendível e compreensível ao serem lidas. Para o exemplo de temperatura, exemplos de regras de inferência são dados por:

IF valorTemperatura IS calor THEN diminuirTemperaturaAr

IF valorTemperatura IS agradável THEN manterTemperaturaAr

O conjunto resultante após a aplicação destas regras apresenta todos os resultados considerados adequados conforme o valor de entrada recebido. Desta forma, é necessário apenas 'escolher' um valor como resultado final. Isto é feito no processo de defuzzificação.

2.3.1.3 Defuzzificação

Esta fase consiste no processo inverso ao realizado na fuzzificação. Ou seja, o resultado obtido é convertido de linguístico para numérico (*crisp*) novamente e, com isto, pode ser aplicado no sistema operante. Existem diversos métodos que podem ser aplicados para realizar a defuzzificação: COG, MeOM, RCOM, FOM (DERNONCOURT, 2013; LEEKWIJCK; KERRE, 1999). Em (KLIR; YUAN, 1995) são apresentadas as vantagens e desvantagens de alguns deles.

2.3.2 Considerações

Como mencionado em (MARRO et al., 2010), a incerteza e imprecisão dos dados são um desafio enfrentado por sistemas, principalmente para sistemas inteligentes. Quando isto ocorre, inferir um resultado sem o uso do método adequado, pode resultar em resultados imprecisos e até inadequados. Além destes tipos de sistemas (com dados faltantes), a lógica fuzzy é capaz de contribuir em sistemas altamente complexos e difíceis de modelar ou ainda, sistemas que trabalhem com aspectos como o comportamento humano.

Além destes fatores, a tomada de decisão em problemas reais frequentemente faz uso de dados de sensores, os quais podem ser falhos (incompletos) e/ou imprecisos. A utilização de lógica fuzzy permite que o resultado final não seja afetado devido a estes fatores. Por outro lado, a lógica fuzzy por si só, não permite que seja contemplada multi-

objetividade sem a priorização de um ou outro aspecto. No entanto, existe uma extensão da lógica fuzzy baseada no uso de *lattices* (WINTER, 2007) que atende estes requisitos e é mais adequada à não linearearidade (comum quando há multiobjetividade): \mathcal{L} -fuzzy.

2.4 TEORIA DOS LATTICES E \mathcal{L} -FUZZY

Para ilustração de um problema multiobjetivo, Goguen (GOGUEN, 1967) utilizou como exemplo o caso de uma dona de casa. Esta, ao ir ao supermercado, busca selecionar entre as diversas opções ofertadas, um pacote de alimentos que contemple diversos critérios de optimalidade como: custo, valor nutricional, qualidade e variedade. Neste problema, não é adequado dizer que um produto com maior valor nutricional é melhor que um de maior variedade, por exemplo. Desta forma, com o propósito de prover uma solução para problemas como este, considerando propriedades incomparáveis sem dar maior importância a uma ou outra, em 1967 Goguen (GOGUEN, 1967) generalizou o conceito Fuzzy para conjuntos e relações \mathcal{L} -fuzzy para um arbitrário *Brouwerian Lattice* completo ao invés do intervalo de unidade $[0,1]$.

Um '*Lattice*' (\mathcal{L}) (WINTER, 2007), também conhecido como *reticulado* é um elemento matemático que corresponde a um conjunto finito (P) parcialmente ordenado (*poset*), sendo que quaisquer dois elementos deste conjunto podem ser relacionados um ao outro através da relação binária "menor ou igual", possuindo um supremo e um ínfimo. Um *poset* será chamado linear se todos seus elementos obedecerem a propriedade de $x \leq y$ ou $y \leq x$. Elementos deste tipo de conjunto satisfazem as propriedades de:

- Reflexividade: $x \leq x$ para todo $x \in P$
- Transitividade: se $x \leq y$ e $y \leq z$ então $x \leq z$, para todo $x, y, z \in P$
- Antissimetria: se $x \leq y$ e $y \leq x$ então $x=y$, para todo $x, y, z \in P$

Considerando M um subconjunto de P , as seguintes propriedades podem ser identificadas:

- Cota superior (*Upper bound*): elemento x de M onde todos elementos de M são menores ou iguais a x .

$$M \in P, x \in M, \text{ para todo } y \text{ em } M: y \leq x$$

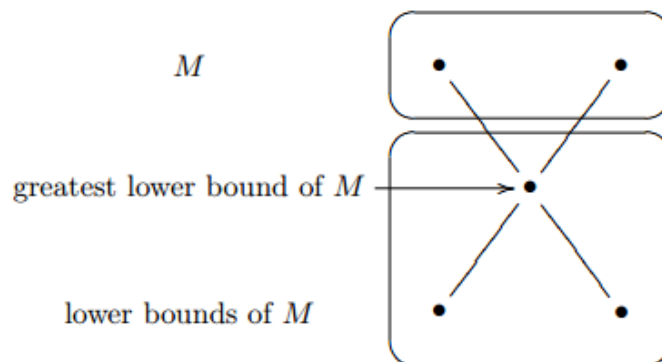
- Cota inferior (*Lower bound*): elemento x de M onde todos elementos de M são maiores ou iguais a x .

$$M \in P, x \in M, \text{ para todo } y \text{ em } M: y \geq x$$

- Maior elemento (*Greatest element*): é um elemento de M , que também é um *upper bound* em M . Este elemento, quando existente, é único.
- Menor elemento (*Least element*): é um elemento de M , que também é um *lower bound* em M . Este elemento, quando existente, é único.
- Supremo (*Least upper bound*): menor elemento do conjunto de *upper bounds* de M .
- Ínfimo (*Greatest lower bound*): maior elemento do conjunto de *lower bounds* de M .

Na Figura 2.5, observe que para cada elemento do conjunto de cota superior, os demais elementos são ou menores ou iguais a ele.

Figura 2.5 – Cota superior, inferior e ínfimo de M .



Fonte: Adaptado de (WINTER, 2007).

Neste caso o conjunto de *lower bounds* apresenta 3 elementos, sendo que um deles é maior que todos os outros deste conjunto. Este elemento é o ínfimo, descrito anteriormente.

Quando todo subconjunto de um *lattice* apresenta valores *crisp*, supremo ($1 = \sup$) e ínfimo ($0 = \inf$), este é considerado um 'reticulado completo'.

Um *lattice* pode também ser definido como uma estrutura algébrica onde são aplicadas operações de união (\cup) e conjunção (\cap), que correspondem respectivamente ao supremo e ínfimo entre um par de elementos x e y pertencentes ao *poset*. Por exemplo, se aplicarmos estas operações aos valores $x = 0.5$ e $y = 0.8$ no intervalo de unidade $[0, 1]$ para do conjunto M de números reais, teremos os seguintes resultados:

$$x \cup y = 0.8 \text{ e } x \cap y = 0.5$$

Em um exemplo mais genérico onde $x \leq y$, teremos:

$$x \cup y = y \text{ e } x \cap y = x$$

Baseado nestas propriedades e na seção 2.3, que apresentou a lógica fuzzy, a próxima seção introduz o conceito das relações \mathcal{L} -fuzzy.

2.4.1 Conjuntos e Relações \mathcal{L} -Fuzzy

Como descrito na seção anterior, um *lattice* (\mathcal{L}) sempre apresentará algum valor, seja máximo (1) ou mínimo (0). Isto pode não ocorrer na lógica fuzzy em geral, ou seja, simplesmente aplicando lógica fuzzy é possível que existam casos onde não se obtenha um valor para o grau de pertinência em uma ou mais funções de pertinência. Pela aplicação do conceito de *lattice* a lógica fuzzy, será possível que sempre se tenha um resultado, mesmo que este seja algo como “sem alterações” ou então “manter configuração”.

Existem diferentes tipos de *lattices* (WINTER, 2007). Neste trabalho foram utilizados os conceitos dos chamados *Brouwerian lattices* ou *Heyting algebras*, que são identificados quando a função característica de uma relação ou conjunto provê como resultado *crisp* o menor elemento 0 ou maior elemento 1 do intervalo de unidade $[0,1]$. Além das funções de união e intersecção existentes em *lattices* em geral, *Brouwerian lattices* introduzem a possibilidade do uso de complemento e negação (conversão). Como demonstrado na definição 2.24 em (WINTER, 2007), neste tipo de *lattice*, existe um pseudocomplemento para cada par de elementos:

“Considere um *lattice* \mathcal{L} e $x, y \in \mathcal{L}$. Um pseudocomplemento relativo $y:x$ de x em y é um elemento em que

$$u \leq y:x \Leftrightarrow x \cap u \leq y$$

para todo $u \in \mathcal{L}$. Um *lattice* no qual para cada par de elementos existe o pseudocomplemento relativo é chamado um *Brouwerian lattice* ou uma *Heyting algebra*.” (WINTER, 2007)

Um conjunto \mathcal{L} -fuzzy (*L-set*) A em um conjunto X é dado pela função $A : X \rightarrow \mathcal{L}$. O que significa que, dado um conjunto X , o conjunto \mathcal{L} -fuzzy A será dado por uma função que apresentará o grau (optimalidade, coerência, certeza...) do elemento em X . Devido a característica de completude do *lattice*, sempre existirá um valor para cada elemento do conjunto, ou seja, todo elemento apresentará um grau mesmo que este seja 0.

Uma relação \mathcal{L} -fuzzy (*L-rel*), por sua vez, consiste em uma função entre dois conjuntos não vazios (A e B) para \mathcal{L} . Para estas relações, podem ser aplicadas operações como: união, conjunção, conversão e composição. Por exemplo, dadas as relações $R: A \rightarrow B$ e $S: B \rightarrow C$, as seguintes operações são identificadas:

- $(R \cap S)_{(x,y)} := R_{(x,y)} \cap S_{(x,y)}$

A intersecção sobre dois pares (x,y) em R e S , será um novo par contendo o menor valor identificado em cada elemento.

- $(R \cup S)_{(x,y)} := R_{(x,y)} \cup S_{(x,y)}$,

A união sobre dois pares (x,y) em R e S , será um novo par contendo o maior valor identificado em cada elemento.

- $R'_{(x,y)} := R_{(y,x)}$,

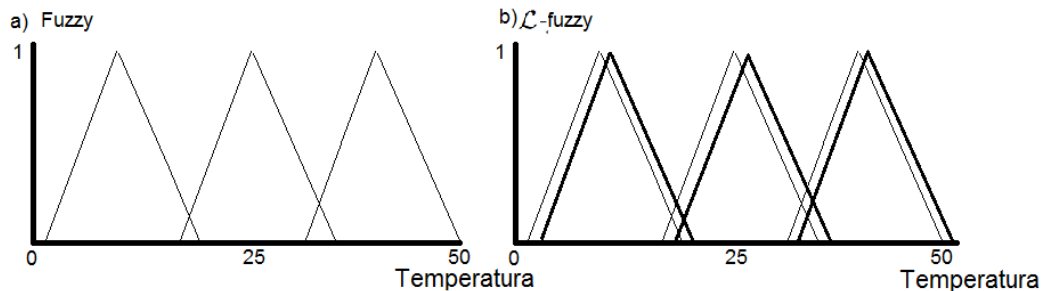
A conversão sobre um par (x,y) em R , será o mesmo par, porém com a posição de x e y invertida.

- $(R \circ S)_{(x,z)} := \bigcup_{y \in B} (R_{(x,y)} \cap S_{(y,z)})$

A composição sobre dois pares em R e S , os quais apresentam respectivamente apenas o segundo e primeiro elemento equivalentes (y , neste caso), será um novo par contendo o primeiro elemento de R e o segundo elemento de S .

A Figura 2.6 ilustra exemplos de funções de pertinência fuzzy e \mathcal{L} -fuzzy para a variável linguística temperatura. No gráfico fuzzy (a), dado o valor ótimo 25, é gerada a função de pertinência 'adequada' em relação a este valor. Porém, se considerarmos um problema multiobjetivo, e houvesse um segundo valor ótimo 27, esta função já não estaria representando este valor adequadamente, pois o valor 27, apresenta um grau de pertinência (μ) inferior a 1. Este problema é resolvido com o uso de \mathcal{L} -fuzzy, como ilustrado em (b).

Figura 2.6 – Representação de variáveis de pertinência Fuzzy X \mathcal{L} -Fuzzy.



Fonte: Próprio autor.

2.4.1.1 Controlador \mathcal{L} -fuzzy

A teoria \mathcal{L} -fuzzy pode ser utilizada para tomada de decisão multiobjetiva por meio do desenvolvimento de controladores capazes de raciocinar sobre problemas multiobjetivos. O raciocínio realizado por um controlador \mathcal{L} -fuzzy (LFC) gera um resultado final *crisp*, o qual indica uma solução possível para o problema correspondendo, portanto, a decisão tomada. Por exemplo, para o problema multiobjetivo citado anteriormente (Figura 2.6), em um cenário onde busque-se controlar a temperatura do ambiente baseado na preferência de dois usuários (25° e 27°), um LFC tomaria uma decisão gerando um resultado que busque contemplar ambas preferências sem priorizar uma ou outra.

Além de apresentar a teoria matemática sobre relações \mathcal{L} -fuzzy, Winter (WINTER, 2007) apresenta um capítulo direcionado especificamente a controladores neste contexto. Neste capítulo, Winter resume um controlador \mathcal{L} -fuzzy em uma operação da seguinte forma:

$$C = L^{in\sim}; R; L^{out} \quad (2.6)$$

onde C representa o *core* (núcleo do controlador), $L^{in\sim}$ corresponde a um *L-rel* com valores de entrada para o qual é aplicada uma operação de conversão, R representa as regras de inferência, $;$ é uma função de composição L^{out} corresponde a um *L-rel* com valores de saída.

A teoria matemática desenvolvida em (WINTER, 2007) permite o raciocínio sobre aspectos como correctude dada uma especificação, propriedades de segurança, além de outras. Um dos maiores problemas abordados para o controlador é garantir sua totalidade, ou seja, o controlador deve produzir um valor de saída para cada possível valor de entrada. Por meio de cálculos baseados nesta teoria, as propriedades do controlador foram provadas. É importante ressaltar que esta dissertação buscou apresentar brevemente estes conceitos com o objetivo de justificar o uso deste tipo de controlador. Porém, para maior compreensão da teoria matemática que este se baseia, sugerimos uma leitura mais profunda a (WINTER, 2007), e também a (GRÄTZER, 2002), (MUNARETTO, 2005), (BIRKHOFF, 1940) e (BARR; WELLS, 1990).

O controlador utiliza as notações de categorias Goguen permitindo que valores não lineares sejam utilizados, ou seja, não se limita ao intervalo $[0,1]$. A metodologia Mandami (MARRO et al., 2010) é utilizada apresentando etapas que definem as variáveis linguísticas, processo de fuzzificação, base de regras, inferência e defuzzificação. Para compreensão destas etapas, utilizaremos o exemplo de um controlador da temperatura de um ar condicionado, o qual apresenta como saída um valor que corresponde a configuração deste.

2.4.1.1.1 Variáveis linguísticas

Um controlador fuzzy é formulado por noções abstratas que representam palavras comuns da linguagem humana (“consumo alto”, “muito quente”). Estas palavras, no contexto fuzzy, são conhecidas como variáveis linguísticas e se aplicam também a conjuntos \mathcal{L} -fuzzy. Com o objetivo de identificar o conjunto de variáveis linguísticas pertinente ao controlador, Winter (WINTER, 2007) apresenta a técnica da aplicação de ‘modificadores’. Ou seja, primeiramente é identificado o termo que traz uma noção básica de uma entidade abstrata como, por exemplo, ‘temperatura agradável’. Para este termo, a partir de um valor considerado ‘ótimo’, é identificado um conjunto de valores que apresentam algum grau de similaridade a ele. Para tal, pode ser utilizada uma classe de modificadores que modela o

conceito de ‘igualdade aproximada’. Esta classe identifica demais valores que apresentam um grau de pertinência superior a 0 em relação ao valor ótimo.

Um exemplo da fundamentação deste conceito pode ser dado pelo exemplo da temperatura identificada por um ser humano, ou seja, um ser humano geralmente não consegue identificar a diferença entre 0 e 1° ou 35 e 36°. Isto ocorre porque estes valores são aproximadamente iguais. No entanto, essa igualdade não é transitiva ou então todos os valores seriam considerados iguais. Com isto, o conceito de grau representa adequadamente esta aproximação. Para obtenção deste conjunto, Winter utiliza a *Lukasiewicz t-norm* (2.7):

$$\min(1, \max(0, a - (b * |x \sim y|))) \quad (2.7)$$

onde x e y correspondem a valores contidos no intervalo finito dos valores possíveis para a variável e a e b são coeficientes definidos pelo especialista que expressam uma proximidade do valor x em relação a y . Por exemplo, se aplicada a definição (2.7) para a temperatura 36° (y) em relação a 35° (x), dados os coeficientes $a=1.2$ e $b=0.4$, o grau de similaridade de 36° em relação a 35° será 0.8.

Na teoria fuzzy, este termo inicial pode ser modificado por advérbios como ‘muito’, ‘pouco’, etc... De uma forma abstrata, estes advérbios podem ser vistos como modificadores linguísticos (intensificadores ou redutores), os quais podem ser computacionalmente representados por funções que mapeiam um conjunto \mathcal{L} -fuzzy para outro. Os conceitos de *upper* e *lower bound* podem ser utilizados para identificação dos modificadores ‘menor que’ e ‘maior que’. Por exemplo, considerando M como o conjunto identificado como ‘média’, são definidos os modificadores redutores e intensificadores: “extremamente”, “muito”, “mais ou menos” e “pouco”. Para tal, as seguintes inclusões devem ser respeitadas:

$$\text{extremamente}(M) \subseteq \text{muito}(M) \subseteq M \subseteq \text{maisoumenos}(M) \subseteq \text{pouco}(M) \quad (2.8)$$

Para implementação destes modificadores, Winter (WINTER, 2007) apresenta a utilização de um parâmetro deslocador (u), o qual é implementado de uma maneira que possibilita a definição de seu valor por um especialista. Sendo assim, a partir do conjunto inicial (L -rel média - Q_0), o grau de um elemento (x) nos demais L -rels corresponde ao valor do grau identificado na L -rel anterior na posição x , porém somado (intensificadores) (2.9) ou subtraído (redutores) (2.10) ao deslocador u .

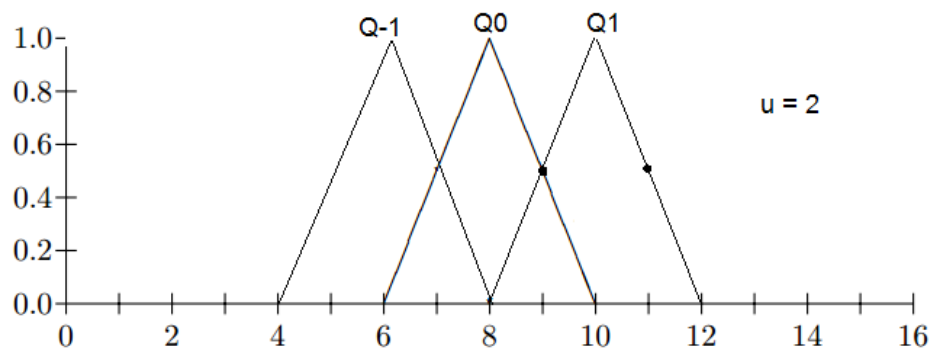
$$Q_1(x) = Q_0(x + u); Q_2(x) = Q_1(x + u)... \quad (2.9)$$

$$Q_{-1}(x) = Q_0(x - u); Q_{-2}(x) = Q_{-1}(x - u)... \quad (2.10)$$

A Figura 2.7 ilustra a aplicação dos modificadores Q_1 e Q_{-1} para a variável Q_0 , considerando como deslocador (u), o valor 2. Ou seja, dado um valor ótimo 8, primeiramente

foi encontrado o conjunto que representa a igualdade aproximada a este valor (Q_0). A partir disso, com a aplicação do deslocador 2, foram identificados dois modificadores: um intensificador (Q_1) e um redutor (Q_{-1}). Para identificação do grau dos elementos em Q_1 , foram identificados os valores existentes em Q_0 com um deslocamento de 2 elementos. Por exemplo, o grau de pertinência existente no valor $x=9$ em Q_0 , será aplicado ao valor $x=11$ ($9+2$) em Q_1 .

Figura 2.7 – Modificadores deslocados com base em u .



Fonte: Próprio autor.

A partir da definição de todos conjuntos \mathcal{L} -fuzzy de uma variável linguística, o conjunto abstrato desta variável é modelado em uma Goguen categoria por meio da união disjunta destes conjuntos.

2.4.1.1.2 Fuzzificação

Em (WINTER, 2007), a fuzzificação consiste de uma operação $F(x)$ a qual mapeia cada valor de entrada x para um conjunto \mathcal{L} -fuzzy. Os valores de entrada são interpretados por elementos relacionais ou pontos. Pontos são um tipo especial de conjuntos \mathcal{L} -fuzzy (WINTER, 2007), portanto, o método de fuzzificação mais fácil é a própria identidade do elemento. Portanto, no exemplo utilizado, se for obtido 20° como valor de entrada, a função de fuzzificação retornará este mesmo valor.

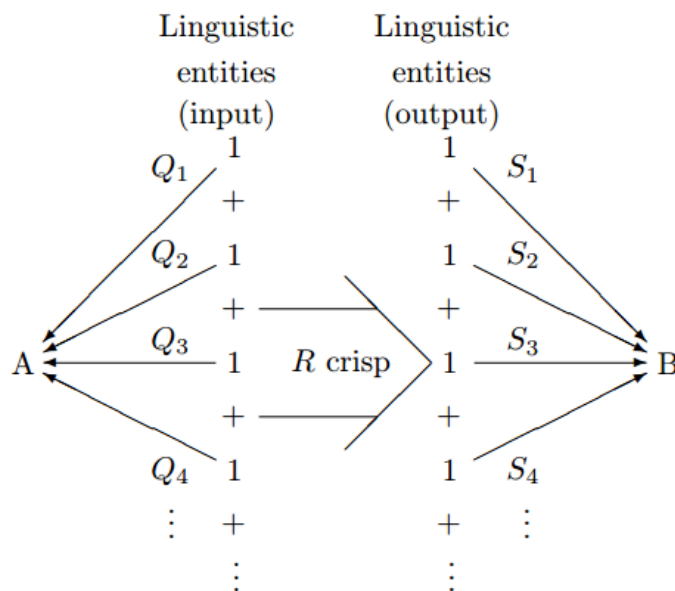
2.4.1.1.3 Base de regras

Para a base de regras, Winter (WINTER, 2007) utiliza expressões condicionais fuzzy (2.5). Desta forma, uma regra de inferência pode ser representada por meio de um par de entidades linguísticas como, por exemplo:

IF temperatura is PoucoAlta, THEN configuracao = ReduzirPouco

Sendo assim, a base de regras, que pode ser constituída de uma lista de regras de controle, pode ser descrita por uma relação R (ver definição 2.6) entre os conjuntos de entidades linguísticas. Assim como nos conjuntos definidos para uma variável linguística, em uma categoria Goguen arbitrária, estes conjuntos são modelados por meio da uniao disjunta das entidades. Pela combinação do conjunto com a base de regras e os conjuntos com as variáveis linguísticas de entrada e saída, obtemos o termo relacional para o núcleo do controlador. A Figura 2.8 ilustra este núcleo, baseado na definição (2.6):

Figura 2.8 – Núcleo do controlador \mathcal{L} -fuzzy.



Fonte: (WINTER, 2007).

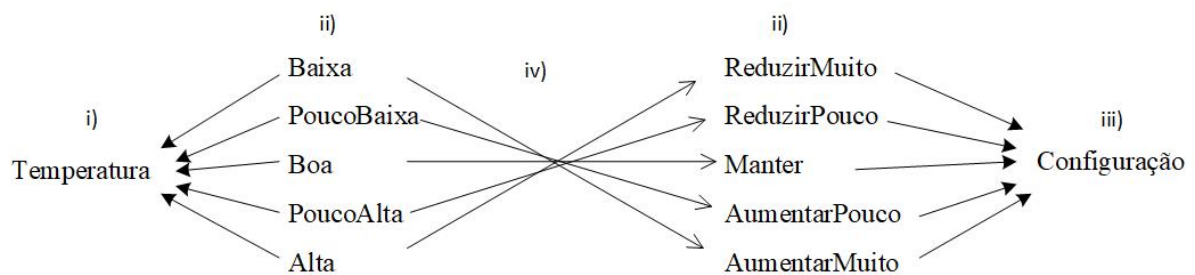
Na teoria das relações, descreve-se um subconjunto A através de uma relação $M: I \rightarrow A$, onde I é uma unidade da categoria relacional correspondente. Na Figura (2.6), o valor o A corresponde a variável linguística de entrada (L^{in}) e B a de saída (L^{out}), as quais são compostas pela união disjunta das unidades representadas pelo numero 1. Q_s e S_s , são as entidades linguísticas (funções de pertinência) correspondentes. R corresponde a relação aplicada entre tais conjuntos (regras de inferência). A Figura 2.9 exemplifica o uso deste núcleo.

2.4.1.1.4 Módulo de Inferência e Defuzzificação

O módulo de inferência descreve em qual grau uma regra do controlador será ativada para um dado valor de entrada (Figura 2.9-i). Ou seja, este módulo identifica como

o resultado da fuzzificação (Figura 2.9-ii) será combinado ao núcleo do controlador. Neste ponto, qualquer valor que não atende as regras de controle (Figura 2.9-iv) definidas para o contexto do controlador será eliminado. Portanto, este método corresponde parcialmente a tomada de decisão do controlador, pois nele são identificados todos os valores possíveis de saída para cada possível valor recebido na entrada. Por exemplo, se para o valor de entrada (temperatura) for recebido um valor alto, o qual apresenta graus de pertinência relevantes para as funções “poucoAlta” e “alta”, de acordo com as regras de inferência, os valores contidos nas funções “reduzirMuito” e “reduzirPouco” são possíveis valores de saída (Figura 2.10).

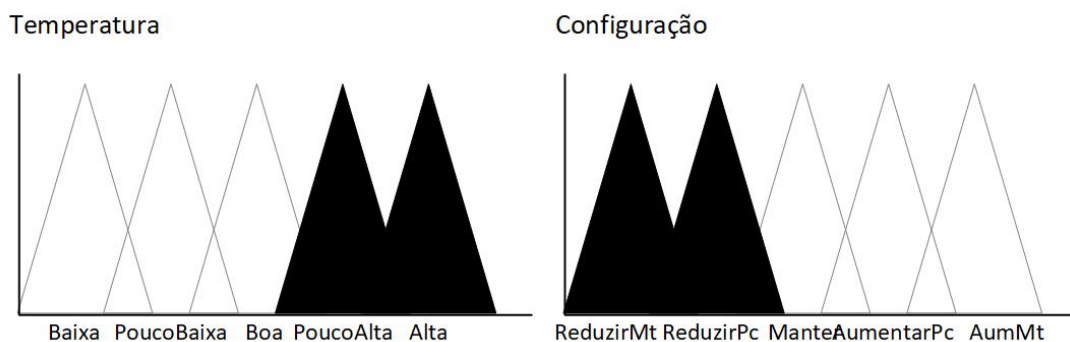
Figura 2.9 – Exemplo de núcleo do controlador.



Fonte: Próprio autor.

Se o valor de entrada estiver apenas na função de pertinência “baixa”, os possíveis valores de saída são os valores pertencentes a função “aumentarMuito” e assim por diante. Ou seja, o conjunto resultante do módulo de decisão corresponde a todas soluções possíveis e adequadas ao problema, sendo necessário apenas ‘escolher’ uma destas soluções como resultado final. Esta escolha é feita na etapa de defuzzificação.

Figura 2.10 – Exemplo de aplicação das regras de inferência.

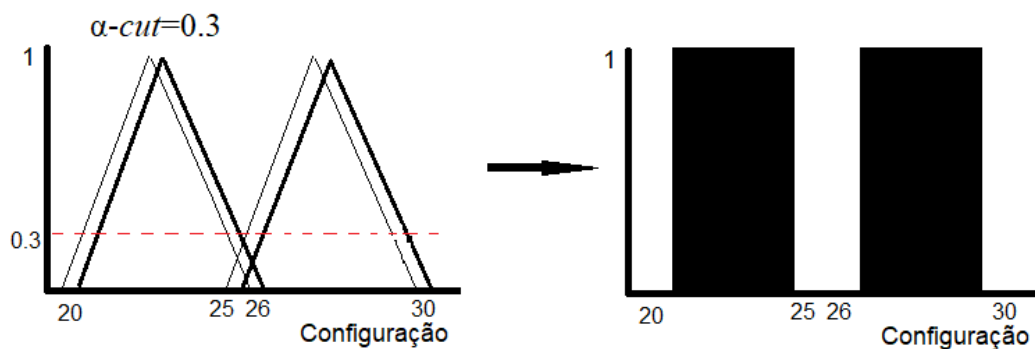


Fonte: Próprio autor.

Em um processo Fuzzy, a defuzzificação transforma o valor de saída novamente em um valor *crisp*. Semelhante a fuzzificação onde se aplica uma operação $F(x)$, na defuzzificação é aplicada uma operação $D(x)$, a qual mapeia uma relação \mathcal{L} -fuzzy para *crisp*. Como

exemplificado na seção 2.3.1.3, existem diversos métodos para defuzzificação. Entretanto, geralmente estes métodos estão associados ao intervalo de unidade $[0,1]$ ou valores lineares. Devido a este fato, estes métodos não são aplicáveis ao controlador definido por (WINTER, 2007), que trabalha com *lattice* e valores não lineares. Com isto, a escolha do método ‘*D*’ depende geralmente do *lattice* L que é utilizado para modelar o grau de pertinência no controlador. No entanto, existe pelo menos uma classe de métodos que pode ser utilizada para todos *lattices*: métodos derivados de uma operação ‘ α -cut’. Nesta operação, dado um valor mínimo para ativação de uma regra, percorre-se o conjunto \mathcal{L} -fuzzy resultante e atribui-se o máximo (supremo) valor (1) para todos elementos iguais ou acima do α -cut definido, e o mínimo (ínfimo) valor para todos elementos abaixo. Por exemplo, dado o valor α -cut=0.3, um elemento do conjunto final com grau 0.1 será convertido para 0 e um elemento com grau 0.8, será convertido para 1. Considerando que a inferência resultou nas funções “reduzirMuito” e “reduzirPouco”, todos valores com grau abaixo de 0.3, serão eliminados do conjunto final (Figura 2.11). Após a aplicação da operação α -cut, o conjunto restante é o conjunto *Pareto*, sendo que qualquer elemento deste conjunto pode ser escolhido como resultado final.

Figura 2.11 – Defuzzificação usando operação ‘ α -cut’.



Fonte: Próprio autor.

O uso de *lattice* (pares) possibilita que a origem do resultado final seja conhecida e quanto cada critério influenciou no resultado final. Em metodologias que fazem uso de pesos como, por exemplo, ANP (MOHANTY et al., 2005) esta informação seria perdida pois o resultado final é calculado conforme o peso do critério. Por exemplo, dados o valor 0.2 para o primeiro critério e 0.6 para o segundo, supondo que ambos pesos sejam 0.5, obtém-se o resultado final de 0.4 ($0.5 \cdot 0.2 + 0.5 \cdot 0.6 = 0.4$). Porém, o mesmo resultado seria obtido se ambos valores fossem 0.4. Em um conjunto final com dois valores 0.4 ($\{0.4, 0.4\}$) pode ser preferível escolher o valor onde ambos critérios contribuíram igualmente, porém esta informação já foi perdida. Como consequência, estas metodologias permitem que um critério com valor alto compense um que apresenta valor muito baixo, mas nem sempre isto é o desejado. Na metodologia de (WINTER, 2007), o uso de *lat*-

tice possibilita 'escolher' o valor mais adequado do conjunto final $\{(0.2, 0.6), (0.4, 0.4)\}$. No entanto, caso o especialista julgue necessário, esta metodologia ainda permite que seja dada prioridade a alguma variável. Isto pode ser feito por meio das regras de inferência como, por exemplo: `If consumo is alto AND temperatura is adequado, then configuracao = aumentarTemperatura.`

Outra característica do controlador é que ele é implementado independente do valor de entrada e a decisão é tomada o mais tarde possível (no processo de defuzzificação). Ou seja, a mudança dos valores de entrada não resulta no total recálculo dos valores no controlador. Entretanto, antes de iniciar sua utilização, é necessária a realização de diversos testes com o objetivo de configurá-lo corretamente e obter os resultados de saída esperados.

Conforme o propósito da aplicação, o resultado do LFC pode ser aplicado com o objetivo de controlar o ambiente por meio da manipulação da situação deste. Uma possível maneira de realizar esta manipulação é por meio das chamadas *ações automatizadas*.

2.5 AÇÃO AUTOMATIZADA

Ações automatizadas (Aa) são ações realizadas pelo sistema ou realizadas diretamente pelos dispositivos por meio de suas funcionalidades. Estas funcionalidades correspondem a serviços que, normalmente, são implementados pelo fabricante do dispositivo e estão disponíveis para consumo pela aplicação. Um serviço (Se) pode ser representado pela definição (2.11) (MACHADO et al., 2017):

$$Se : (\{I\}, \{O\}) \quad (2.11)$$

onde $\{I\}$ corresponde ao conjunto de entidades de entrada e $\{O\}$ a entidades de saída. Estas entidades são entidades do contexto.

Portanto, de acordo com o modelo definido em (MACHADO et al., 2017), uma ação automatizada (Aa) (Figura 2.1-d) é disparada após a identificação de uma ou mais situações $\{Sp\}$ que são precondições para sua execução; é composta por um conjunto de serviços $\{Se\}$ (funcionalidades dos dispositivos), o qual deve ser organizado em ordem lógica de consumo; e resulta em um conjunto de eventos internos $\{E\}$. Desta forma, (MACHADO et al., 2017) caracteriza uma ação automatizada da seguinte forma:

$$Aa : (nome, \{Sp\}, \{Se\}, \{E\}) \quad (2.12)$$

Uma ação automatizada é executada com o objetivo de realizar um ajuste que visa manipular uma situação de interesse no ambiente, buscando alcançar a situação desejada.

Contudo, não há como garantir que este objetivo será alcançado, pois diversos fatores podem impedir isto como, no exemplo de um sistema que controla a temperatura do ambiente, pode ocorrer uma súbita mudança na temperatura causada pela abertura de uma porta e conseqüente entrada de ar quente na sala. No entanto, indiferente do alcance ou não do objetivo, a execução de uma ação automatizada sempre gera um evento interno. Este evento pode resultar no término da situação atual ou, até mesmo, no início de uma nova situação.

O próximo capítulo apresenta trabalhos relacionados aos diferentes conceitos aplicados neste trabalho.

3 TRABALHOS RELACIONADOS

Não foram encontrados na literatura, trabalhos que contemplem todos os aspectos do trabalho desenvolvido nesta pesquisa (sensibilidade ao contexto e situação, tomada de decisão multiobjetiva e \mathcal{L} -fuzzy). Desta forma, este capítulo tem como objetivo apresentar trabalhos relacionados aos diferentes tópicos abordados neste trabalho, ou seja, sensibilidade ao contexto e/ou situação, o problema multiobjetivo relacionado a busca por conforto e eficiência energética em residências, lógica fuzzy e tomada de decisão multiobjetiva fuzzy.

3.1 SENSIBILIDADE AO CONTEXTO E/OU SITUAÇÃO EM AMI

Diante da ausência de uma metodologia em engenharia de software que seja diretamente adequada para o *design* de sistemas Ami, os autores em (CORONATO; PIETRO, 2010) propõem um modelo contendo um processo de especificação para o *design* de atividades e artefatos Ami. O modelo auxilia *designers* na produção de especificações verificáveis e não ambíguas para aplicações Ami, incluindo a especificação de requisitos, entidades, atividades e ferramentas necessárias para seu desenvolvimento. Para identificação do contexto, o modelo faz uso de uma ontologia em seu estágio inicial. Este modelo não apresenta tomada de decisão correspondendo, portanto, somente aos primeiros módulos do modelo definido neste trabalho.

(FENG; TENG; TAN, 2009) apresentam um modelo computacional sensível ao contexto para suporte a tomada de decisão. O modelo visa reduzir a carga humana utilizada no comando e controle de sistemas na classificação de eventos, recomendação de ações e tomada de decisão proativa. Com base na definição de Endsley para SA, o modelo compartilha a informação provida pelo SA utilizando-a de maneiras diferentes de acordo com o contexto de cada usuário (agente). Para identificar qual informação é relevante a situação de cada usuário, o modelo utiliza a informação do contexto deste, ou seja, o modelo recupera a informação de SA necessária a cada usuário com base em seu contexto para suporte a decisão deste. Como ferramenta para o suporte a tomada de decisão, os autores utilizaram uma ferramenta de inferência baseada em regras: DROOL. O modelo de (FENG; TENG; TAN, 2009) é adequado para sistemas que ofereçam suporte a tomada de decisão e não a tomada desta em si. Ou seja, apesar de reduzir a carga do usuário, eles ainda interagem bastante com este. Além disso, apesar do modelo possibilitar a tomada de decisão para múltiplos usuários de acordo com os objetivos destes, isto não é feito de uma forma multiobjetiva, ou seja, que exija que todos os objetivos sejam considerados para que seja tomada uma única decisão. Em resumo, apesar de apresentar características simila-

res às deste trabalho devido a utilização de sensibilidade ao contexto/situação e tomada de decisão, o modelo de (FENG; TENG; TAN, 2009) é aplicado a contextos diferentes do contexto abordado neste trabalho, tendo um objetivo diferente.

(BOUSDEKIS et al., 2016) propõem um modelo probabilístico para aprimorar tomada de decisão proativa controlada por eventos. Com o objetivo de prover recomendações adequadas, o modelo considera sensibilidade ao contexto quando há incerteza sobre os valores dos elementos contextuais. A tomada de decisão é disparada pelo evento preditivo que contém a função de probabilidade de distribuição do evento indesejado. O método de decisão ou provê a ação ideal a ser tomada acompanhada do tempo ideal para sua execução, ou então, provê o tempo adequado para execução de uma ação predeterminada. O modelo faz uso de Redes Bayesianas, as quais provêm a probabilidade que um custo específico é válido, dado um contexto específico. Em partes, o modelo se assemelha ao desenvolvido neste trabalho, pois dispara a tomada de decisão a partir da predição de um evento indesejado conforme o contexto. No entanto ele não apresenta um módulo para execução automática da ação, nem considera multiobjetividade.

3.2 SITUATION AWARENESS E LÓGICA FUZZY

O raciocínio fuzzy pode ser aplicado nos três níveis de SA. Em sistemas SA a etapa de percepção ocorre, geralmente, por meio de sensores ou de outras aplicações que transmitam dados, os quais podem ser incompletos ou não apresentar total precisão. A compreensão dos dados baseia-se em uma base de conhecimento, a qual pode ser uma base de conhecimento fuzzy definida por meio das funções de pertinência. E por fim, na etapa de projeção os resultados podem ser inferidos com a aplicação de regras de inferência fuzzy.

Em seu trabalho, (KOKAR; MATHEUS; BACLAWSKI, 2009) citam o desafio na identificação de uma situação existente entre um cão e gato, para a qual necessita-se identificar se eles estão brincando (*playing*) ou brigando (*threatening*). Os autores demonstram a contribuição de lógica fuzzy em SA por meio de seu uso na identificação da situação atual, a qual é vaga e incerta. Além disso, eles propõem uma ontologia que serve como a base de conhecimento necessária para a geração das regras de inferência fuzzy. A seguir são apresentados outros trabalhos onde foi identificada a integração destas áreas.

(NADERPOUR; LU, 2012) consideraram a aplicabilidade de conceitos SA para segurança no controle de sistemas complexos. Sistemas de supervisionamento de segurança têm tido um forte crescimento no grau de automação e complexidade, enquanto que o número de operadores destes tem diminuído. Se por um lado, operadores têm ganho quanto a maior facilidade na performance de suas atividades, por outro, eles enfrentam maior dificuldade quanto ao vasto número de dados gerados e interpretação correta da

informação. Os autores descrevem a implementação de uma abordagem baseada em Sistemas Especialistas (ES) com o objetivo de aprimorar SA em ambientes complexos. ES são sistemas que contemplam um vasto conhecimento sobre um domínio e são capazes de fazer inferências e chegar a uma conclusão baseado nos valores de entrada, da mesma maneira (ou quase) que um especialista humano no assunto faria. ES fazem uso de lógica fuzzy com o objetivo de simular o raciocínio humano. De acordo com os autores, agir rápido, obter informação clara e objetiva, prever/antecipar situação e focar em prioridades são aspectos altamente importantes para garantia da segurança em ambientes complexos. No modelo definido, baseado em regras e base de conhecimento especialista, fuzzy é capaz de inferir situações de risco as quais são reportadas para o operador. No estudo de caso apresentado, o qual busca identificar situações de risco em um processo industrial de síntese de etilbenzeno, utilizando fuzzy foram definidas as variáveis linguísticas, as funções de pertinência (para identificação dos níveis de risco) e, finalmente, a base de regras fuzzy. A matriz resultante foi utilizada para construir o sistema especialista em análise de riscos fuzzy responsável pela emissão de alertas pertinentes.

Em (MAIO et al., 2012) os autores ressaltam que aplicações SA, comumente, fazem grande uso de sensores, os quais dificilmente transmitem dados perfeitamente ou com exatidão. Nestas aplicações, diversas redes e tipos de sensores transmitem dados sobre os mais variados elementos e isso resulta, muitas vezes, em 'muitos dados' mas 'pouca informação'. O artigo visou definir um *framework* que explorasse a disponibilidade de dados semânticos de sensores visando identificar situações relevantes em diversos ambientes dinâmicos. Os autores aplicaram controles fuzzy em dados semânticos visando prever situações emergentes e a seguir foi aplicado raciocínio fuzzy sobre os dados obtidos da situação.

(DANIELLO; LOIA; ORCIUOLI, 2015) destacam a importância da qualidade nos dados utilizados em SA para que seja garantido o reconhecimento correto da situação. Uma aplicação não tem como saber se a informação obtida é baseada em dados parciais obtidos dos sensores, se o sensor não enviou os dados ou estes foram enviados, mas eram inválidos. Pensando neste aspecto, (DANIELLO; LOIA; ORCIUOLI, 2015) sugere que o grau de confiabilidade dos dados sendo utilizados deve ser informado ao mecanismo de identificação de situação. Com isto, o artigo propôs uma abordagem baseada em Tecnologias Semânticas e *Fuzzy Consensus Model* (FCM). FCM foi utilizado visando prover *observações* confiáveis por meio do processamento de dados obtidos de sensores redundantes. Estas *observações* são geradas pelos sensores e foram representadas por meio de termos linguísticos fuzzy definidos em um conjunto. Para seleção de uma *observação* considerada confiável em meio às outras de diferentes sensores do mesmo grupo, foi utilizado um processo conhecido como *Group Decision Making* (GDM), o qual consiste em um problema a resolver, um conjunto solução com as alternativas possíveis e um grupo de dois ou mais especialistas que expressam sua opinião sobre as alternativas, as quais são

mapeadas em um vetor com graus de preferência.

Por sua vez, (JONES et al., 2010) descrevem um modelo cognitivo de SA utilizando mapas cognitivos fuzzy (*Fuzzy Cognitive Maps* - FCM) englobando os três níveis de SA descritos por Endsley. O modelo foi nomeado SA-FCM e representa um 'modelo naturalístico computacional de tomada de decisão'. Os autores identificaram que diversas abordagens tradicionais de modelagem cognitiva não consideram SA como um fator cognitivo e algumas outras, mais recentes, modelavam apenas o nível 1 (percepção) de SA. Considerando que SA é identificado como um dos principais fatores que contribuem na qualidade do processo de decisão, os níveis 2 e 3 de SA também devem ser considerados para modelar tomada de decisão de uma maneira que reflita a realidade. Neste contexto, SA-FCM foi projetado com base em requisitos existentes em operações de exércitos, mais precisamente, o modelo se baseou em informações que um soldado utilizaria para tomada de decisão em uma batalha. O modelo proposto é capaz de representar decisões em tempo real (ou aproximado) por meio da eficiente compreensão e projeção do cenário, de maneira semelhante às decisões que seriam tomadas por um ser humano.

3.3 TOMADA DE DECISÃO MULTIOBJETIVA FUZZY

(GOVINDAN; KHODAVERDI; JAFARIAN, 2013) apresentam uma abordagem FMCDM com o objetivo de auxiliar empresas na formação de uma cadeia sustentável de fornecedores. A metodologia proposta auxilia na tomada de decisão na seleção de fornecedores que melhor se encaixem simultaneamente nos critérios ambientais, econômicos e sociais. Para atingir este objetivo, além de raciocínio fuzzy, os autores utilizam o algoritmo TOPSIS para medir a performance de sustentabilidade de cada fornecedor.

(CHEN; TAN, 1994) apresentam técnicas para manipulação de problemas multicritérios fuzzy de tomada de decisão baseado na teoria dos conjuntos vagos. Isto é feito pelo uso de funções que identificam quanto uma alternativa satisfaz determinado critério e quanto ela não o satisfaz em um conjunto vago. Este trabalho foi estendido em (HONG; CHOI, 2000), onde foram providenciadas novas funções que medem o grau de precisão nos graus de pertinência de cada alternativa relacionada a um conjunto de critérios, fornecendo informação adicional sobre as alternativas. Ambos trabalhos fazem uso de pesos para os critérios.

Além do controlador proposto em (WINTER, 2007), não encontramos nenhum trabalho que realize a tomada de decisão utilizando \mathcal{L} -fuzzy.

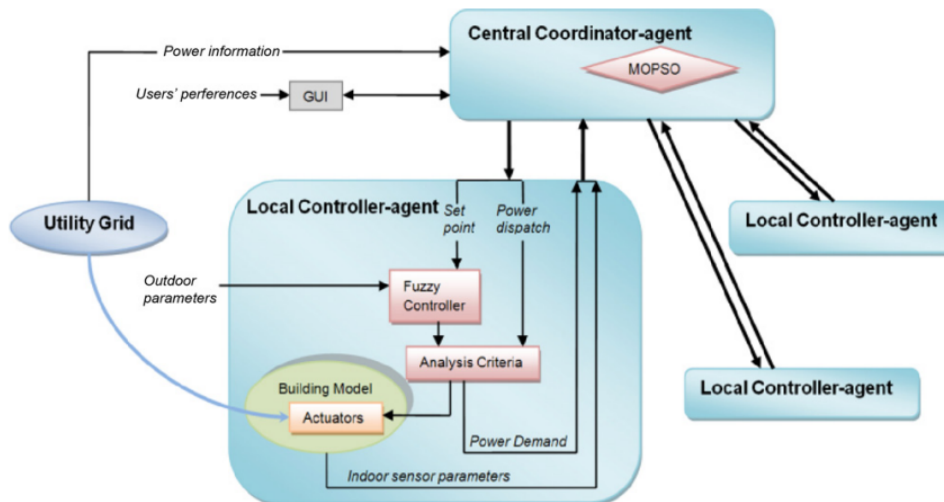
3.4 PROBLEMA MULTIOBJETIVO: CONFORTO X ENERGIA

A busca pela solução do problema multiobjetivo relacionado ao conforto e eficiência energética em residências é uma área que apresenta um vasto número de trabalhos. Por exemplo, (WANG; YANG; WANG, 2010) visaram desenhar um sistema de controle inteligente que visa maximizar o conforto do usuário minimizando o consumo energético. Para atingir este objetivo, foi desenvolvido um sistema multi agentes que gerencia o controle da temperatura, iluminação e ventilação do ambiente, sendo que o usuário pode definir o valor mínimo e máximo para cada um destes agentes. O sistema é composto de múltiplos controladores fuzzy, os quais são utilizados para obter diferentes demandas de conforto baseado em cada um dos agentes definidos. A saída de cada um destes controladores, é o valor de potência necessário para manter o conforto de acordo com o agente. Além disso, para otimizar o sistema de controle, é aplicado o algoritmo PSO (*particle swarm optimization*), que otimiza resultados, geralmente, em conflito. Os autores definem que o conforto do usuário é obtido considerando o valor de erro encontrado entre o valor medido no sensor e o valor definido pelo usuário como adequado.

Em um trabalho posterior, (YANG; WANG, 2012) apresentam um *framework* de controle (Figura 3.1) baseado em multi agentes para aplicações para prédios inteligentes, o qual também objetiva gerenciar o consumo de energia e conforto dos ocupantes de uma residência. Os autores visam alcançar estes objetivos por meio do controle do conforto térmico (variável: temperatura, controlado por meio dos sistemas de resfriamento/aquecimento), conforto visual (variável: iluminação, controlada pelo sistema elétrico) e qualidade do ar no ambiente interno (variável: concentração de dióxido de carbono, controlada pelo sistema de ventilação). Foi considerado o parâmetro 'conforto' como sendo um conforto em geral, o qual é obtido com a agregação de todas estas variáveis. O *framework* é composto de um agente coordenador central e diversos agentes controladores locais, os quais fazem uso de controladores fuzzy para calcular a potência necessária para manter um nível alto de controle em cada variável. O agente coordenador central se comunica com o usuário e demais controladores agentes locais que lhe enviam informações sobre os parâmetros, sendo ele responsável pela tomada de decisão para controle geral do prédio. A partir destas informações, o agente coordenador calcula o conforto do ambiente e então tenta maximizar o conforto e diminuir o consumo de energia. O algoritmo MOPSO (*multi-objective particle swarm optimization*) é utilizado para ajudar a tomar decisões adequadas para o consumo de energia. Foram comparados os conjuntos de soluções Ótimo Pareto obtidos com o uso deste algoritmo aos obtidos por meio do uso da metodologia *weighted aggregation* (a qual transforma múltiplos objetivos em um único objetivo por meio de pesos definidos pelo usuário para as variáveis). Como resultado desta comparação, foi identificado que os conjuntos apresentavam valores semelhantes.

Semelhante ao trabalho desenvolvido em (WANG; YANG; WANG, 2010), (SHAIKH

Figura 3.1 – *Framework* de controle do sistema multi-agentes.



Fonte: (YANG; WANG, 2012).

et al., 2016) apresentam um sistema coordenador multi objetivos para gerência de consumo de energia e do conforto residencial, o qual utiliza como método de otimização o algoritmo MOGA (*evolutionary multi-objective genetic algorithm*). No entanto, além dos parâmetros conforto visual, térmico e qualidade do ar, (SHAIKH et al., 2016) introduzem mais um parâmetro considerado importante no conforto em um ambiente: a umidade do ar. Para cada um dos parâmetros definidos, o sistema proposto apresenta agentes coordenadores periféricos e, além destes, um coordenador *master* que os coordena. Cada parâmetro recebe um peso de importância e valores mínimo e máximo que correspondem a valores ideais para que o ambiente seja considerado confortável são definidos pelo usuário para estes. Considerando estes valores, o coordenador *master* aplica o algoritmo MOGA com o objetivo de maximizar cada valor o mais rápido possível e ajustar o consumo de energia. MOGA é um método de otimização evolucionário baseado em algoritmos genéticos que otimiza e atualiza os valores de configuração (geralmente em conflito) do agente *master* e periféricos em cada passo, adicionando inteligência ao sistema. Ele é o responsável pelo alcance de um resultado que satisfaça os critérios de conforto e controle do consumo de energia, seus valores de saída. Assim quando um coordenador periférico não consegue atingir sua meta, o *master* oferecerá mais potência para que ele alcance este valor o mais rápido possível. Com o objetivo de superar o problema de não linearidade no processo de controle, controladores fuzzy (FC) são utilizados para identificar a demanda de potência de cada parâmetro. A entrada dos dados nestes controladores é processada em forma de erro (valor real - valor esperado), assim é possível que o FC não se limite ao uso de uma unidade e tem-se uma saída (*output*) comum para todos os parâmetros. Nestes controladores, dois valores de saída são obtidos: (a) potência, que é dado pela soma da potência de cada parâmetro, e (b) conforto: para definição final do conforto.

Por sua vez, (YU et al., 2015) destacam eficiência energética e conforto térmico como sendo dois critérios conflitantes que precisam ser incorporados ainda no *design* de construções. Desta forma, o *design* pode ser considerado um problema de otimização multiobjetiva. (YU et al., 2015) justificam que uma arquitetura que atenda estes aspectos é complexa, existindo a necessidade do uso de serviços que ajudem a encontrar as melhores opções de *design*. Com isto, os autores propõem um modelo para otimização multiobjetiva que visa ajudar *designers* de construções a definir o *design* adequado no estágio inicial deste processo. A partir de determinadas condições, o modelo proposto auxilia o *designer* a tomar decisões efetivas e apropriadas, bem como evitar potenciais erros por meio da escolha das variáveis mais adequadas que otimizam o conforto térmico e eficiência energética do ambiente. Para atingir este objetivo, o modelo faz combinação de uma ANN (*artificial neural network*) e do algoritmo genético multiobjetivo NSGA-II (*multi-objective evolutionary algorithm*). Este algoritmo é utilizado para otimizar a ANN com o objetivo de aprimorar a acurácia da predição feita por esta. Uma rede GA-BP mais precisa é gerada para predição de consumo de energia e conforto térmico sendo esta treinada e testada utilizando resultados simulados.

3.5 CONSIDERAÇÕES

Uma das diferenças entre os controladores implementados nos trabalhos citados nas seções 3.4 e 3.3 e o desenvolvido neste trabalho, são os valores ideais para as variáveis. No controlador deste trabalho, um único valor é definido como ideal (*optimal*). No entanto, este valor será fuzzificado sendo obtidos valores semelhantes a ele em um certo grau. Em (WANG; YANG; WANG, 2010) e (SHAIKH et al., 2016), por exemplo, os valores considerados ideais para conforto são intervalos definidos pelo usuário. Porém, estes intervalos não são fuzzificados, ou seja, se o usuário definir o intervalo 22-25° para temperatura, o valor 25.5°, por exemplo, está fora da temperatura ideal obtendo um valor de erro, o que não reflete o raciocínio humano que seria incapaz de identificar esta diferença considerando 25.5° um valor ideal. Além destes intervalos, uma característica importante encontrada nos trabalhos apresentados é o uso de metodologias que exigem a definição de pesos de importância para os critérios. O peso dado a um critério afeta o resultado final obtido. Em alguns trabalhos, o peso é definido pelo especialista e em outros existe uma função que os calcula aleatoriamente. Este trabalho não faz uso de pesos, pois visa a não priorização dos critérios, ou seja, considera que nenhum parâmetro pode ser considerado mais importante que o outro.

Os trabalhos apresentados consistem em controladores definidos para um domínio específico (conforto e eficiência energética), não sendo possível ou não tendo sido mencionada a possibilidade de sua adaptação para outros domínios. O modelo desenvolvido

nesse trabalho pode ser aplicado em qualquer contexto multiobjetivo, não se limitando somente ao estudo de caso utilizado neste trabalho.

Outra característica destes trabalhos, é que eles são multiagentes, ou seja, os sistemas desenvolvidos utilizam controladores diferentes (um para cada parâmetro), obtendo diferentes resultados de conforto e demanda de potência, para só então aplicar um algoritmo que otimize o valor de saída. Com a aplicação das propriedades de *lattice* na tomada de decisão, é possível que um único controlador processe as diferentes variáveis de entrada.

Seguindo esta linha de raciocínio, apesar destes trabalhos contemplarem a tomada de decisão multiobjetiva, pouca atenção tem sido prestada a outros aspectos como a detecção da situação que desencadeia a necessidade dessa tomada de decisão. Ou seja, eles não identificam como/se serão realizados o raciocínio sobre o contexto de interesse e a identificação da situação de interesse que dispara a tomada de decisão. Consequentemente, eles correspondem somente ao módulo de decisão definido no modelo proposto neste trabalho. De forma similar, alguns deles não identificam como a decisão tomada será aplicada no contexto.

Em contrapartida, os trabalhos sensíveis ao contexto identificados na seção 3.1 oferecem suporte a tomada de decisão, porém não trabalham com problemas multiobjetivos. Desta forma, com o propósito de aplicar qualquer um dos trabalhos apresentados em um sistema automático para tomada de decisão multiobjetiva em ambientes inteligentes, estes trabalhos ainda precisariam ser complementados. Outra alternativa é o desenvolvimento de novas metodologias, como o modelo desenvolvido neste trabalho, o qual é apresentado na próxima seção.

4 MODELO PROPOSTO

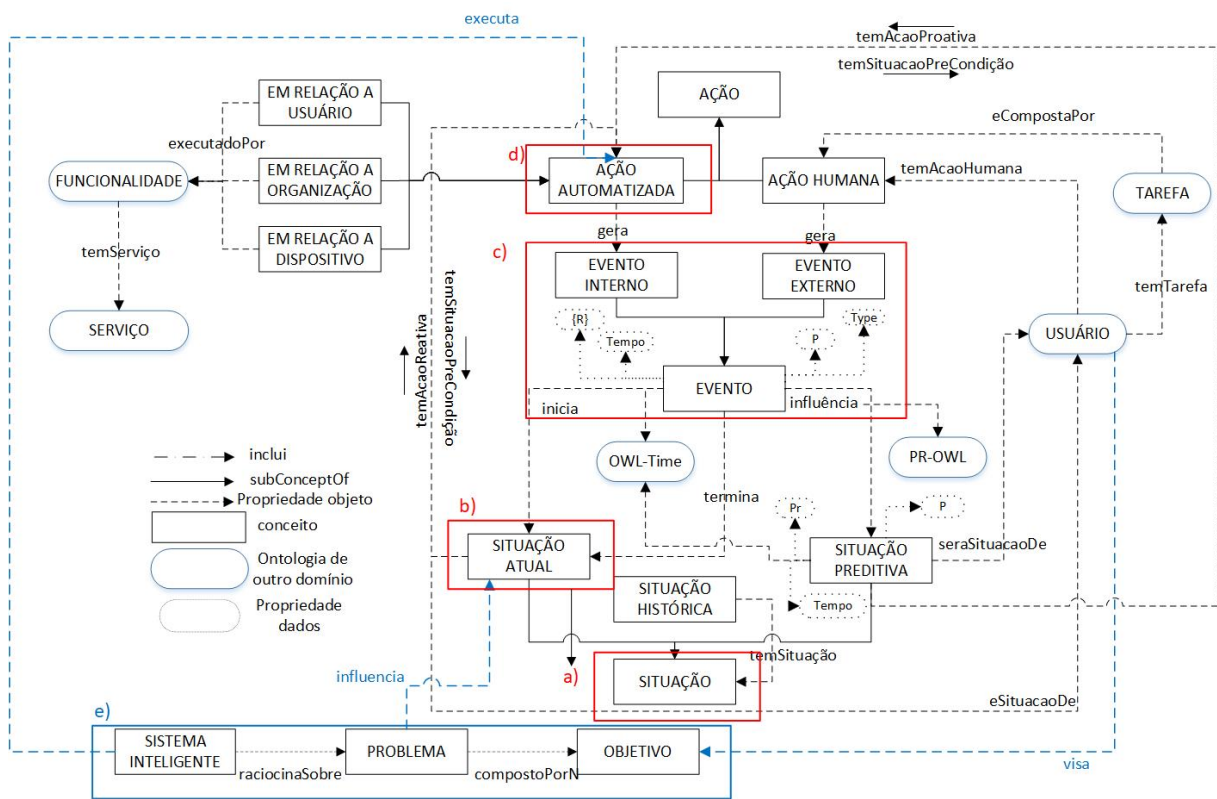
De acordo com (AARTS; WICHERT, 2009), Ambientes Inteligentes (Aml) consistem de ambientes tecnologicamente sensíveis e adaptáveis que respondem as ações de pessoas e objetos para atender suas necessidades. Além da importância de planejar as decisões e ações a serem tomadas em Aml, (RAMOS; AUGUSTO; SHAPIRO, 2008) destacam a necessidade de sistemas Aml conhecerem o ambiente do usuário com o objetivo de melhor atender suas necessidades e interagir inteligentemente com este. Este capítulo apresenta o modelo definido neste trabalho, o qual considera como objeto de interesse problemas com caráter multiobjetivo em Ambientes Inteligentes.

Neste trabalho foi desenvolvido um modelo sensível ao contexto que busca dar aos sistemas a capacidade de tomar uma decisão multiobjetiva baseada no conhecimento de um especialista com o objetivo de executar uma ação que visa modificar a situação atual do ambiente. Para definição deste modelo, foram extraídas do modelo conceitual definido em (MACHADO et al., 2017) as definições de situação, evento, ação automatizada e a relação entre estes elementos (descritos no capítulo 2). Além destas características, como pode ser visto na Figura 4.1 (e), foram adicionados os conceitos de sistema inteligente, problema, objetivo e suas respectivas relações semânticas, necessárias à solução de problemas multiobjetivos em Aml. O conceito de sistema inteligente foi adicionado visando complementar o modelo com a definição de contexto de (ABOWD et al., 1999), a qual inclui o próprio usuário (já existente no modelo de (MACHADO et al., 2017)) e a aplicação.

Este modelo foi desenvolvido com o objetivo de ser aplicado no desenvolvimento de sistemas para ambientes inteligentes, sendo genérico em relação ao domínio do sistema e dinâmico quanto ao número de objetivos contemplados. Sua aplicação é indicada em sistemas que buscam reproduzir o raciocínio humano na solução de problemas multiobjetivos com objetivos, em geral, conflitantes. Exemplos, são sistemas que controlam a temperatura do ambiente de acordo com a preferência de mais de um usuário; sistemas que auxiliam o usuário a realizar mais de uma tarefa ao mesmo tempo ou sistemas que buscam o conforto do usuário ao mesmo tempo que se preocupam com a eficiência energética. No caso deste último exemplo, podemos observar que o modelo pode ser aplicado não apenas como uma maneira de oferecer uma melhor experiência ao usuário, mas também para contribuir com a sustentabilidade do ambiente. Este cenário é discutido em um estudo de caso neste trabalho.

Como mencionado anteriormente, normalmente, decisões neste tipo de ambiente são tomadas visando manipular ou tentar reproduzir aspectos incertos como o raciocínio humano apresentando, frequentemente, características não lineares. As tecnologias aplicadas para o desenvolvimento de cada módulo podem variar conforme a necessidade da aplicação. No entanto, estas tecnologias devem ser capazes de implementar os modelos

Figura 4.1 – Inclusão no modelo conceitual dos conceitos e relações semânticas necessárias à solução de problemas multiobjetivos em Aml



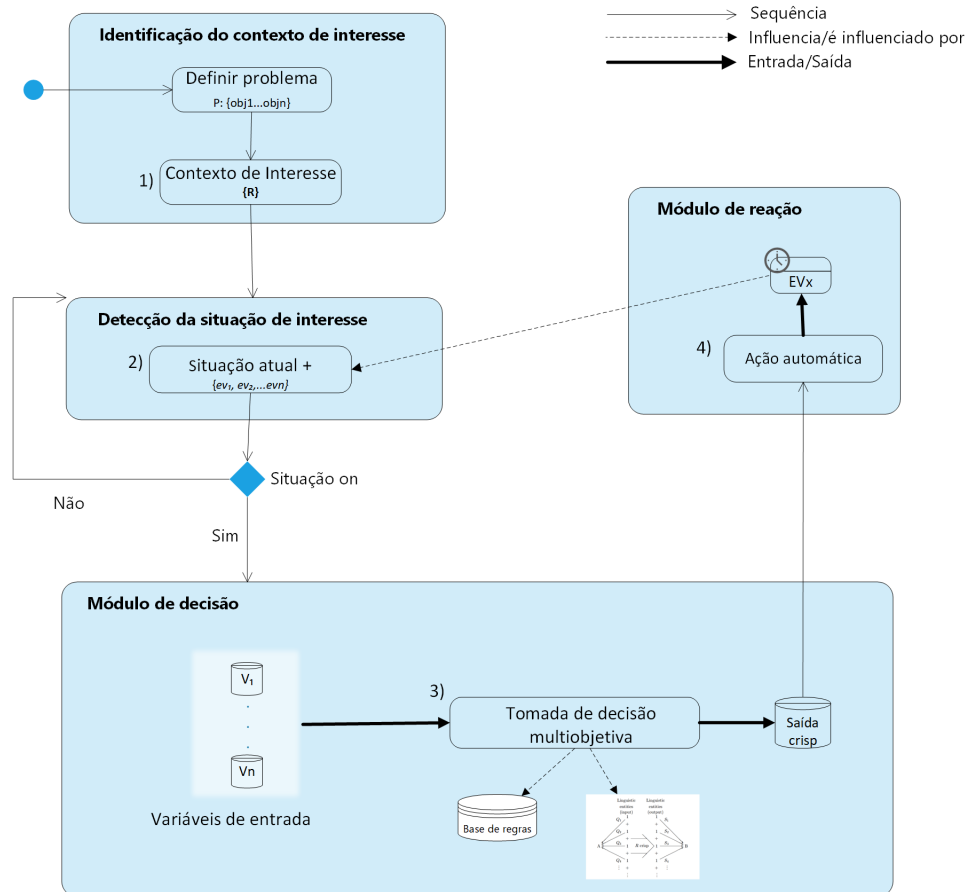
Fonte: Adaptado de (MACHADO et al., 2017).

definidos para cada um destes módulos. Desta forma, este modelo visa atender alguns critérios chave, sendo estes:

- **multiobjetividade:** o foco do modelo está em problemas que apresentam mais de um objetivo a ser atendido, sendo estes objetivos conflitantes entre si, para os quais não deve haver priorização. Isto exige o uso de tecnologias para tomada de decisão multiobjetiva;
- **não-linearidade:** objetivos diferentes, normalmente, apresentarão intervalos diferentes e podem não estar restritos ao intervalo de unidade $[0, 1]$. Além disso, o raciocínio humano é considerado não linear;
- **reação em tempo real:** com o objetivo de adaptar o ambiente conforme a necessidade, o sistema deve utilizar tecnologias que permitam agir em tempo real. Isto pode incluir, por exemplo, o uso de dados fornecidos por sensores e tecnologias que permitam um raciocínio rápido;
- **incerteza:** além da tentativa de reprodução do raciocínio humano, o qual é incerto, sensores utilizados nestes sistemas podem apresentar dados incertos e imprecisos.

O modelo desenvolvido, representado na Figura 4.2, é composto de 4 módulos distintos: 1) identificação do contexto de interesse, 2) detecção da situação de interesse, 3) módulo de decisão e 4) módulo de reação.

Figura 4.2 – Modelo



Fonte: Próprio autor.

No módulo de identificação do contexto de interesse, primeiramente, deve ser identificado o problema P para qual a aplicação será desenvolvida, ou seja, deve-se definir qual problema multiobjetivo esta aplicação deve manipular. A identificação deste problema (4.2) é feita por meio de um conjunto contendo todos os objetivos que devem ser considerados. Um objetivo obj (4.1) apresenta um *nome* e o conjunto de variáveis (v) que lhe influenciam. Os valores destas variáveis serão utilizados como entrada no sistema.

$$obj : (nome, \{v\}) \quad (4.1)$$

$$P : \{obj_1, obj_2, \dots, obj_n\} \quad (4.2)$$

As variáveis que influenciam o objetivo variam conforme este objetivo, devendo ser definidas pelo especialista. Por exemplo, como utilizado em (WANG; YANG; WANG, 2010), para o objetivo ‘conforto residencial’, pode ser utilizado o conjunto de variáveis temperatura, iluminação, ventilação:

```
obj : (conforto residencial, {temperatura, iluminação, ventilação})
```

Entretanto, supondo que deseja-se desenvolver uma aplicação para controle da temperatura de um ambiente, a qual vise o ‘conforto térmico’ de dois usuários, podemos considerar que ela apresenta 2 objetivos: ‘conforto térmico usuário 1’ e ‘conforto térmico usuário 2’. Para ambos objetivos, pode ser utilizada a variável ‘temperatura’:

```
obj1 : (conforto térmico 1, {temperatura})
obj2 : (conforto térmico 2, {temperatura})
P : {obj1, obj2}
```

A partir desta definição, pode ser identificado o contexto de interesse para a aplicação. Ou seja, primeiramente são identificadas as entidades (percepção em SA) e correspondentes relações (compreensão em SA) que apresentam relevância para o problema de acordo com os objetivos propostos.

A identificação do contexto de interesse é feita por meio do uso do conceito de relações semânticas $\{R\}$ (2.1) definidas em (MACHADO et al., 2017). Estas relações semânticas devem corresponder a relações entre instâncias de entidades existentes no contexto. Em Aml, frequentemente (mas não necessariamente) as entidades correspondem aos sensores existentes no ambiente. Para o exemplo dado, exemplos de relações semânticas que identificam o contexto de interesse seriam:

```
<ArCondicionado, temSensor, Temperatura>,
<ArCondicionado, estaLocalizado, salaEstar>,
<UsuarioA, estaLocalizado, salaEstar>,
<UsuarioB, estaLocalizado, salaEstar>,
<UsuarioA, temPreferencia, ValorA>,
<UsuarioB, temPreferencia, ValorB>
```

Esta primeira etapa do modelo compreende os níveis de percepção e compreensão definidos em *situation awareness*.

After the definition of the context of interest, it is possible to carry out the second module of our model: detection of the situation of interest. In this module we will identify the so called

A partir da definição do contexto de interesse, é possível avançar para o segundo módulo de nosso modelo: detecção da situação de interesse. Neste módulo identificaremos a chamada ‘situação de interesse’ (S_i). A situação de interesse (4.3) corresponde a situação atual (S_a) do ambiente, porém nela ocorre uma variação no evento de início. A situação de interesse é composta de um conjunto de um ou mais eventos pré-definidos

(*evSet*), os quais identificam um padrão de interesse no domínio. Estes eventos podem indicar, por exemplo, que os valores obtidos dos sensores estão inadequados para o ambiente e não atendem os objetivos estipulados. Com isto, a situação de interesse pode ser enquadrada no nível de percepção em SA, pois é a partir desta situação que é detectada a necessidade de agir com o objetivo de alterar algo no ambiente.

Em problemas multiobjetivos, dificilmente haverá um momento onde os valores de entrada possam ser considerados adequados, pois estes valores podem ser convenientes para um objetivo, mas não para o outro. Portanto, com o propósito de evitar processamento excessivo pelos sistemas, é conveniente encontrar intervalos aceitáveis para estes valores. Uma forma de fazer isto é definir um valor mínimo e máximo para cada variável de entrada. Outra forma é utilizar métodos que, a partir de um valor ótimo informado pelo especialista ou pelo usuário do sistema, identifiquem valores aproximados como, por exemplo, utilizando lógica fuzzy. Ou seja, ao invés de definir que a preferência do *UsuarioA* é *ValorA=23*, define-se um intervalo para preferência: *ValorA={22,23,24}*.

No exemplo anterior, a partir da definição dos intervalos de temperatura adequados para cada usuário, um exemplo de situação de interesse ocorre quando uma alteração recente (dado um tempo de início) na configuração do ar condicionado resultou na detecção de que temperatura atual do ambiente (situação atual) está adequada (dentro do intervalo) para um usuário, porém inadequada (fora do intervalo) para o outro. Esta situação é detectada por meio de um evento (*ev1*) que identifica os valores inadequados para os objetivos. Como os objetivos da aplicação podem diferir, a situação de interesse (4.3) pode ser detectada por um conjunto de eventos diferentes (*evSet*).

evSet: {*ev1*, *ev2*, ..., *evn*}

$$Si : (nome, evSet, Fe) \quad (4.3)$$

Assim como na situação atual, a situação de interesse pode também apresentar um 'fim', o qual é identificado por um novo evento que identifica seu término (*Fe*). Normalmente, espera-se (porém não garante-se) que este evento ocorra após a execução de uma ação automatizada, definida mais adiante neste modelo.

A situação de interesse desencadeia o processo de tomada de decisão, o qual exige como entrada os dados relacionados aos objetivos que foram identificados nesta fase. Ou seja, ocorrendo a detecção de uma situação de interesse, os valores obtidos são enviados para o módulo de decisão (terceiro módulo). Este módulo tem como entrada um número *n* de variáveis, as quais correspondem as variáveis relacionadas aos múltiplos objetivos a serem considerados no ambiente. Com estes valores é realizado o processo de tomada de decisão multiobjetiva e um valor de saída (*crisp*) é retornado, o qual deve ser aplicado ao contexto posteriormente por meio de uma ação automatizada. No exemplo apresentado, este módulo receberia como entrada o valor de temperatura e, com base nas preferências de ambos usuários (objetivos), retorna um valor de saída correspondente a um ajuste a

ser realizado no ar condicionado posteriormente.

Neste trabalho, este módulo é definido utilizando a teoria \mathcal{L} -fuzzy. Por meio de uma base de regras de inferência, o conhecimento de um especialista é utilizado com o objetivo de processar os valores de entrada e obter uma saída *crisp* que corresponde a decisão tomada. A utilização desta metodologia foi escolhida pois ela contempla todos os critérios chave citados anteriormente. Por meio do uso de \mathcal{L} -fuzzy é possível que seja contemplada incerteza e imprecisão dos dados, multiobjetividade, não linearidade, não priorização dos objetivos, além do fato de que esta metodologia possibilita a decisão tardia do valor de saída, evitando a dependência dos critérios e otimizando a performance no processamento. Para implementação deste módulo, este trabalho disponibiliza uma biblioteca para implementação de controladores \mathcal{L} -fuzzy, a qual é descrita na seção seguinte.

No último módulo (módulo de reação), a aplicação no ambiente do valor obtido no módulo de decisão é feita por meio de uma 'ação automatizada' (Aa), a qual utiliza o modelo definido em (MACHADO et al., 2017). Como condição (Sp) para execução desta ação, deve ser identificada uma situação indicando que houve acionamento da situação de interesse e que uma decisão foi tomada após isto. Como resultado da execução desta ação, espera-se a manipulação da situação detectada previamente no ambiente. Isto ocorre, pois a execução de uma ação automatizada gera um evento (E). Consequentemente, existe a possibilidade de que o evento gerado seja o evento que caracteriza o fim da situação de interesse (Figura 4.1). Se isto não ocorrer, a aplicação deve continuar seu processamento visando a manipulação da situação de interesse. No exemplo em questão, dado um valor de saída -2, a ação automatizada definida para o contexto é '*reduzir Temperatura*', e com a execução desta ação espera-se a obtenção de uma temperatura mais agradável a ambos usuários.

Na Figura 4.3 é apresentado o algoritmo a ser considerado com o objetivo de desenvolver um sistema baseado no modelo proposto. Como pode ser visto no passo 1, ele começa pela definição do domínio contendo os múltiplos objetivos a serem considerados no sistema, os quais têm seus valores lidos nos passos 10 e 11. Passos 3 e 14 são relacionados a geração e execução do controlador \mathcal{L} -fuzzy, respectivamente. Observe que o controlador depende apenas do problema (P) para ser gerado. Passo 4 é relacionado ao primeiro módulo do modelo deste trabalho, identificação do contexto de interesse. Passos 6 e 12 se relacionam a situação de interesse. Finalmente, passos 7, 16 e 17 relacionam-se a ação a ser tomada pelo sistema. Como pode ser visto no passo 8, este algoritmo utiliza um laço de repetição com o objetivo de possibilitar que cada iteração detecte uma nova ocorrência da situação de interesse.

Figura 4.3 – Algoritmo para desenvolvimento de sistemas sensíveis ao contexto para MoDM em Aml

```

1  define the set of all objectives of the problem, P
2  {
3      set the lfuzzy controller, lfc(P)
4      find the set of all entities and relations in the context of interest, CI(P)
5  {
6      define the set evSet of all events that start situation of interest, Si(CI)
7      define the services S in the automatized action, Aa;
8      repeat every X minutes
9  {
10     for each obj in P
11         input[] = read value(obj)
12         IF Si(input) is ON
13     {
14         output = execute lfc(input);
15         IF output != null
16             for each S in Aa
17                 execute S;
18     }
19 }
20 }
21 }
22
23

```

Fonte: Próprio autor.

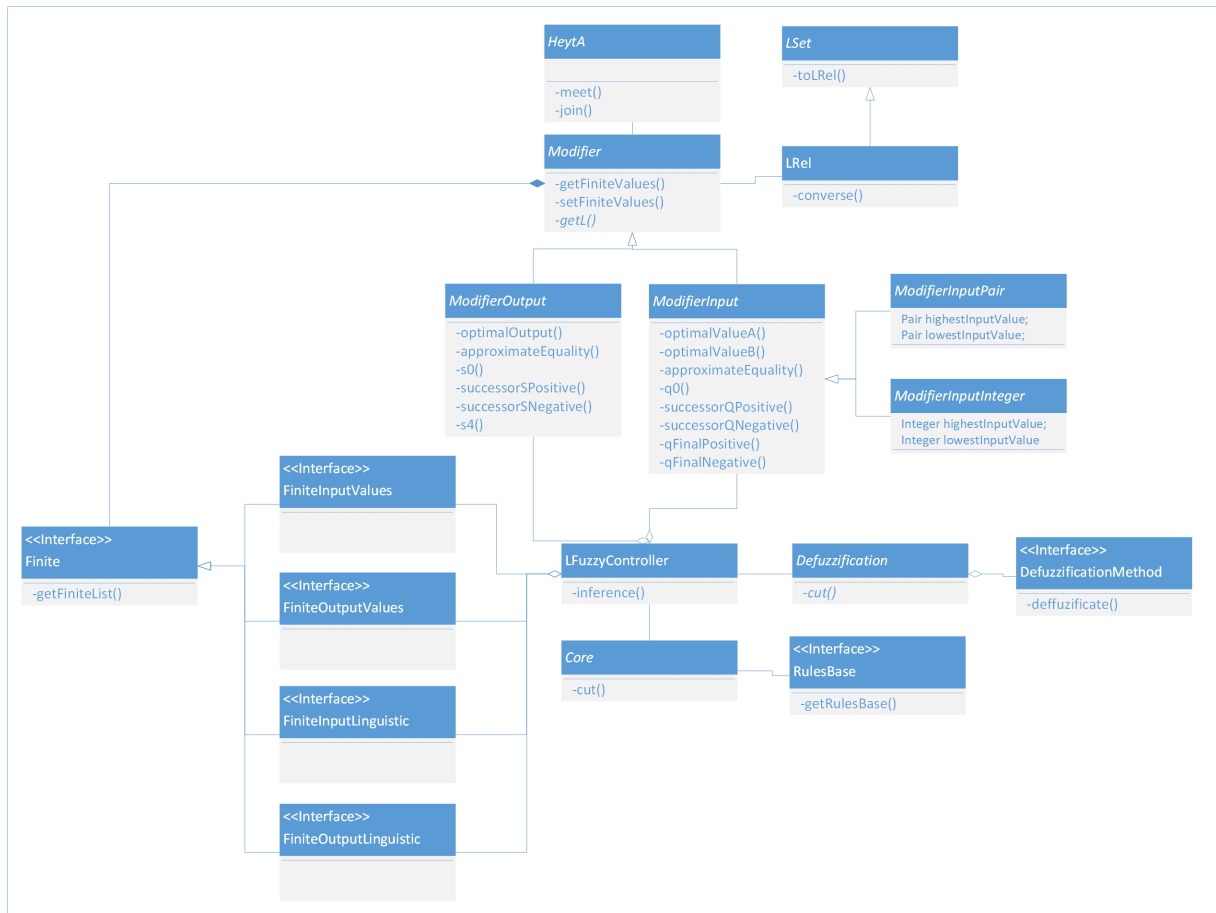
4.1 TOMADA DE DECISÃO

Para o módulo de decisão (Figura 4.2-3), o qual é executado a partir da detecção da situação de interesse, este trabalho desenvolveu uma biblioteca¹ para construção de controladores \mathcal{L} -fuzzy (seção 2.4.1.1) com base nas teorias de *lattice*, *L-sets* e *L-relations* e categorias Goguen, as quais constituem de uma formalização algébrica adequada para trabalhar com relações \mathcal{L} -fuzzy (WINTER, 2007). A teoria \mathcal{L} -fuzzy permite atingir uma solução mais adequada sem priorizar um ou outro aspecto. Com o uso de conceitos e propriedades de um *lattice*, é possível o raciocínio em contextos não lineares e a obtenção de resultados baseados em todos os critérios importantes necessários ao contexto.

Um controlador \mathcal{L} -fuzzy desenvolvido com esta biblioteca tem como entrada n variáveis que correspondem aos múltiplos objetivos sendo considerados no sistema. Ele realiza o processamento destes valores, tomando uma decisão referente a variável de saída que alimenta o módulo sucessor, o módulo de reação que inclui a ação automatizada. A biblioteca, que foi desenvolvida na linguagem Java, é composta de 30 classes e foi inteiramente fundamentada no livro “*Goguen Categories*” (WINTER, 2007). A Figura 4.4 apresenta o diagrama as principais classes da biblioteca.

Como a interpretação de um controlador proposta em (WINTER, 2007) é uma operação dentro de uma categoria Goguen, o autor define propriedades matemáticas (seção 2.4.1.1) sobre o controlador, as quais são provadas por meio do raciocínio na teoria abstrata destas categorias. Isto garante que se tenha uma maior confiabilidade no controlador e, desta maneira, obtemos uma biblioteca apresentando a semântica da lógica fuzzy, aliada

¹<https://github.com/milenetsi/moami>

Figura 4.4 – Diagrama de classes da biblioteca \mathcal{L} -Fuzzy.

Fonte: Próprio autor.

a teoria matemática para raciocínio sobre as noções de corretude desta.

Esta biblioteca foi desenvolvida buscando alcançar o máximo de generalidade possível, podendo assim ser aplicada em qualquer contexto que busque resultados que contemplem multiobjetividade e não linearidade. Para respeitar a multiobjetividade, as classes *HeytA*, *L-rel* e *L-set* fazem uso de parâmetros genéricos possibilitando o uso de multiatributos como pares, triplas, etc. Ou seja, a biblioteca foi implementada de uma maneira escalável para qualquer número arbitrário de objetivos, possibilitando a fácil implementação destes. Por meio da utilização de notações de categorias Goguen, a biblioteca permite que valores não lineares sejam utilizados, ou seja, não se limita ao intervalo $[0, 1]$ dos números reais. No entanto, se desejado, a biblioteca pode ser utilizada também para intervalos lineares, pois estes estão contidos em aspectos não lineares.

Foi utilizada como base a metodologia Mandami (MARRO et al., 2010), a qual consiste de uma base de regras, fuzzificação, módulo de decisão, defuzzificação. Portanto, com a utilização da biblioteca desenvolvida, é possível obter o controlador em poucos passos:

- Identificação das variáveis linguísticas: (seção 2.4.1.1.1) as variáveis linguísticas,

no contexto deste trabalho, são conjuntos \mathcal{L} -fuzzy correspondentes aos objetivos (entrada) a serem contemplados e à saída esperada. Conjuntos \mathcal{L} -fuzzy são implementados nas classes *LSet* e *LRel*: *LSet* é basicamente a classe que armazena um conjunto e seus valores. Esta classe possui dois parâmetros 'A' e 'L'. 'A' corresponde aos valores do conjunto e 'L' a uma *Heyting* álgebra (*HeytA*) relacionada a estes. *LRel*, por sua vez, é uma extensão de *LSet* onde o elemento 'A' corresponde a um par de elementos. Esta classe será aplicada diversas vezes na implementação do controlador.

Classes *Finite* são responsáveis pela definição dos intervalos numéricos a serem usados e *Modifier* pela definição das funções de pertinência para cada variável linguística de entrada (*Qs*) ou saída (*Ss*). A classe *Modifier* é uma classe abstrata que é generalizada em duas outras classes: *ModifierInput* e *ModifierOutput*. Estas classes são responsáveis pelo processamento das variáveis linguísticas, suas funções e graus de pertinência. Nelas os métodos abstratos *getLIn* e *getLOut* devem ser implementados invocando demais métodos modificadores existentes. Estas classes tratam todos os conjuntos como *LRels*. Seus construtores exigem alguns parâmetros para a implementação (ex.: *a, a1, b...u*). Estes parâmetros correspondem a variáveis que devem ser identificadas de acordo com o controlador sendo desenvolvido, ou seja, não existe um cálculo para tais. Os valores devem ser moldados e ajustados até que seja observado o comportamento esperado do gráfico gerado.

Na biblioteca desenvolvida, a união disjunta destes conjuntos corresponde a um método abstrato (*getLIn-ModifierInput* e *getLOut-ModifierOutput*) o qual deve ser implementado gerando todos *L-sets* (funções de pertinência). Após isto, o método deve percorrer a lista finita dos valores da variável associando a entidade linguística correspondente e retornar a união resultante destes a qual é obtida em um único *L-rel*. No exemplo citado, a partir da importação desta biblioteca, o controlador \mathcal{L} -fuzzy é implementado tendo como 'temperatura' e 'configuração' as variáveis linguísticas de entrada e saída respectivamente.

- Fuzzificação: (seção 2.4.1.1.2) neste passo, os dados de entrada relacionados aos objetivos são fuzzificados. A biblioteca não apresenta um método específico para fuzzificação, possibilitando o uso da identidade do dado. Ou seja, no exemplo dado, o próprio valor lido para temperatura pode ser utilizado como dado de entrada. Entretanto, devido ao uso de um tipo genérico para os valores de entrada, é possível que sejam desenvolvidas operações adicionais para tal se desejado.
- Definição da base de regras: (seção 2.4.1.1.3) a base de regras consiste em uma *L-rel* que deve ser implementada de acordo com o conhecimento de um especialista sobre o domínio para o qual o controlador está sendo desenvolvido. Esta base deve ser implementada em uma *L-rel* na classe *RulesBase*. Esta *L-rel* deve conter

pares com elementos *Finite* linguísticos de entrada e de saída, correspondendo às regras definidas. No exemplo, a base de regras é definida por um especialista, que considera a preferência de ambos usuários na sala.

- Módulo de decisão: (seção 2.4.1.1.4) na biblioteca desenvolvida, a classe *Core* apresenta o método *core*, o qual implementa a definição 2.6. O método '*core*' corresponde ao núcleo do processamento. Ele recebe objetos *Modifiers* de entrada e saída, realiza o processamento destes, identifica a relação entre eles e retorna um *LRel* que apresenta o grau de pertinência para cada valor de entrada em cada função de pertinência. O módulo de decisão foi implementado na classe *LFuzzyController*, que é o local onde são definidas as variáveis necessárias para o funcionamento do controlador, bem como os métodos que o executam. No próprio construtor da classe, o método *core* é invocado, identificando um conjunto resultante, porém ainda não considerando valores de entrada nem possíveis restrições no resultado final. Estas restrições são definidas no próximo passo, junto a defuzzificação.
- Defuzzificação: (seção 2.4.1.1.4) *Defuzzification* é a classe abstrata que realizará a defuzzificação do resultado obtido, transformando-o em *crisp* novamente. Esta classe possui um atributo para o objeto que contém o método de defuzzificação (*DefuzzificationMethod*). A defuzzificação é implementada com base em uma operação α -cut, que recebe o *LRel* resultante do processamento e é responsável por eliminar valores que não atingem o grau mínimo esperado de acordo com o valor de entrada. Esta operação encontra-se na interface *DefuzzificationInterface* e precisa ser implementada no desenvolvimento do controlador. Além desta operação, também deve ser definido e implementado um método de defuzzificação (*interface DefuzzificationMethod*) para obtenção do resultado final *crisp*. No exemplo, o valor resultante da defuzzificação é um valor correspondente a nova temperatura a ser aplicada no ar condicionado.

Em resumo, quando uma situação de interesse é detectada, o módulo de decisão é acionado e o controlador fuzzifica o valor de entrada para em seguida aplicar as regras de inferência e identificar o conjunto de saída resultante. Por fim, o método de defuzzificação é aplicado e obtém-se o valor final de saída, o qual é entrada para a ação automatizada.

5 APLICAÇÃO DO MODELO

Com a expansão da tecnologia ocorrida nas últimas décadas, o mundo tem vivenciado a mudança de uma economia baseada em indústria para uma baseada em informação (HILL; FERRIS; MÄRTINSON, 2003). Com o crescimento de tarefas que podem ser realizadas pela internet, muitas pessoas são capazes de trabalhar por meio de dispositivos eletrônicos, deixando de lado os escritórios tradicionais. O número de pessoas e até empresas que optam por *home-office* para seus funcionários tem crescido nos últimos anos (FELSTEAD et al., 2002; JOHNS; GRATTON, 2013; FOWKES; SANDERS et al., 2015). Como estas pessoas passam mais tempo em suas casas, é conveniente a adaptação destes ambientes tanto com o objetivo de prover maior conforto ao usuário e contribuir em sua produtividade, quanto o de reduzir gastos desnecessários. Outro aspecto importante que pode ser buscado nestes ambientes é a sustentabilidade. Com relação ao consumo energético, esta pode ser alcançada com o uso de tecnologias alternativas ou, simplesmente, com o uso mais adequado de equipamentos.

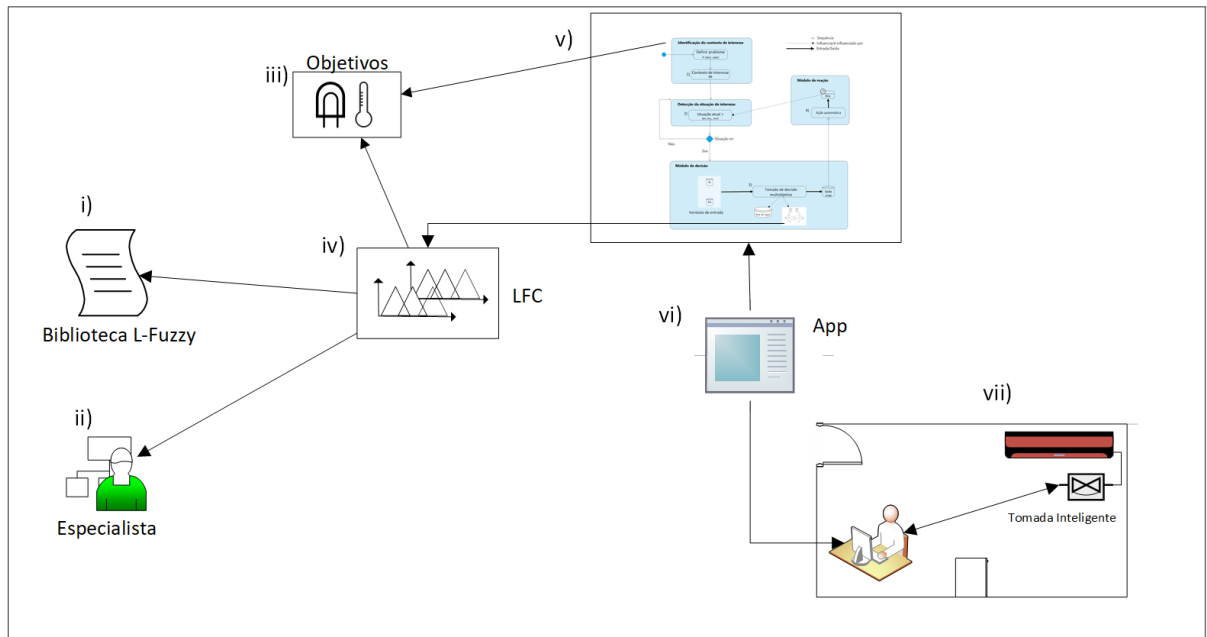
Levando em consideração esses aspectos, este capítulo demonstra como a abordagem desenvolvida nesta dissertação pode ser aplicada em cenários para Ambientes Inteligentes. Com o objetivo de verificar o modelo (v) criado, foi desenvolvido um sistema (vi) para operar no cenário de um *home-office* (vii) para o qual deseja-se manipular o problema multiobjetivo relacionado ao conforto térmico e eficiência energética (iii). Para este sistema, foi implementado um controlador \mathcal{L} -fuzzy (iv) a partir da biblioteca¹ (i) disponibilizada e do conhecimento de um especialista (ii). A Figura 5.1 ilustra as relações existentes entre estes elementos.

O cenário consiste de uma sala em um Aml, a qual é utilizada como escritório de um programador. Considerando que o usuário gasta grande parte de seu tempo neste cômodo e que ele vive em um local de clima quente, com o objetivo de deixar o ambiente termicamente agradável, ele mantém o ar condicionado constantemente ligado. Consequentemente, este ambiente apresenta um elevado consumo de energia, o qual pode ainda apresentar picos de consumo causados por eventos como a abertura de uma porta ou janela, o que permite a entrada de ar quente no ambiente. A princípio, este consumo ocorre de maneira descontrolada resultando em um desperdício de energia, pois não existe um sistema inteligente para gerenciamento desses objetivos.

Para alguns aparelhos de ar condicionado (incluindo o aparelho utilizado neste trabalho), uma das maneiras de evitar o consumo excessivo de energia é por meio do reajuste do valor de temperatura após o alcance da temperatura desejada. Isto é possível, pois estes aparelhos mantêm seu compressor (responsável pelo gasto de energia) ligado até atingir a temperatura desejada. Após isto, eles fazem uso de ventilação (menor consumo),

¹<https://github.com/milenetsi/moami>

Figura 5.1 – Aplicação do modelo no estudo de caso



Fonte: Próprio autor.

desligando o compressor (SORGATO; ROBERTO, 2014). Se a temperatura definida para o ar condicionado for muito inferior/superior a temperatura externa, o compressor será reacionado constantemente buscando alcançar a temperatura desejada. Porém, este esforço excessivo poderia ser evitado a partir de uma pequena elevação/diminuição na temperatura definida para o ar. Desta forma, ele atinge a temperatura desejada, aciona o modo ventilação e mantém uma temperatura razoável no ambiente (sem que o cidadão perceba grande diferença), postergando o novo acionamento do compressor.

Considerando isto, a aplicação desenvolvida neste trabalho visa manter a temperatura necessária para o conforto térmico do usuário, mas também evitar o consumo desnecessário de energia. Considerando que não existe uma fonte de energia alternativa e o aparelho não pode ser simplesmente desligado, por meio do uso da aplicação desenvolvida buscamos adicionar inteligência ao ambiente visando reproduzir decisões que seriam tomadas para o ambiente por um especialista ao tentar controlar a temperatura e reduzir o desperdício de recursos. Este controle é alcançado por meio do reajuste da configuração do ar condicionado que é feito de acordo com a saída obtida no módulo de decisão.

Desta forma, o problema multiobjetivo que exige tomada de decisão no cenário desenvolvido é composto de dois objetivos: (a) conforto térmico do usuário e (b) redução do consumo de energia gerado pelo ar-condicionado, sendo que ambos são considerados importantes neste contexto e, portanto, nenhum deve ser priorizado. É importante mencionar que, dependendo do perfil do usuário, economia de energia deveria ter prioridade sobre conforto térmico, ou o contrário. No entanto, esta opinião varia de pessoa para pessoa e, com o uso de inteligência, ambos objetivos podem ser contemplados de maneira adequada

sem que seja necessário dar prioridade a um ou outro.

Também vale ressaltar que, além do ar condicionado utilizado nestes ambientes, outros equipamentos (computadores, refrigerador, servidor, etc.) também podem apresentar um alto consumo energético. O modelo desenvolvido é dinâmico e pode ser aplicado em qualquer contexto, portanto, poderia contemplá-los. Entretanto, para isto seria necessário um novo estudo de caso incluindo estes elementos em cada módulo do sistema desenvolvido.

5.1 ESTUDO DE CASO: CONTROLE DE CONFORTO TÉRMICO X EFICIÊNCIA ENERGÉTICA

O cenário definido para este estudo de caso consiste de uma sala utilizada como escritório em uma residência. Para resfriamento do ambiente, esta sala contém 1 ar-condicionado *split* 24.000 BTUs (Rheem RB1HW24HP2B), o qual é mantido ligado boa parte do dia. Além do ar-condicionado, na sala existem outros equipamentos utilizados pelo programador: computador, notebook e frigobar. Consideramos que a temperatura do ambiente interno é sensível tanto aos equipamentos eletrônicos quanto às variações de temperatura do ambiente externo, sendo afetada pela abertura da porta de entrada, evento que permite a entrada de ar quente/frio na sala. Para coleta dos dados de temperatura do ambiente e dados relacionados ao ar-condicionado, este está conectado a uma tomada inteligente, a qual se baseia em princípios de IoT (*Internet of Things*) (WEBER; WEBER, 2010) e foi desenvolvida em um projeto de pesquisa² da Universidade Federal de Santa Maria.

Na criação do protótipo da tomada inteligente (SOLIGO, 2017) (Figura 5.2) foi utilizado o micro controlador *Atmega328p* (ATMEL; BOOK,) para controlar o circuito desta. A coleta de dados de corrente foi realizada utilizando o sensor *SCT 013 – 20A*, sensor de corrente não invasivo. Para medir a tensão foi desenvolvido um sensor de tensão, onde foram utilizados quatro diodos, um capacitor e um resistor variável. Assim é possível retificar o sinal variável da rede elétrica em corrente contínua. O resistor variável realiza o micro ajuste do sensor de tensão, o sinal de tensão da saída do secundário do transformador do circuito é retificado para que o micro controlador tenha um sinal de referência da tensão externa. O controle de energia (liga/desliga) é feito por um relé de carga e, a partir do momento que um equipamento é ligado ou desligado, as informações sobre data e hora também são coletadas. Além destes dados, existe um sensor *DHT11*³ acoplado a tomada, o qual é responsável pela coleta de dados de temperatura do ambiente. Por estar fisicamente próxima ao ar condicionado, foi adicionado a esta tomada um microcontrolador

²Projeto de pesquisa nº041766 - Sensibilidade ao Contexto em Sistemas para Ambientes Inteligentes

³<http://www.micropik.com/PDF/dht11.pdf>

capaz de clonar os comandos enviados pelo controle remoto do ar condicionado. Este microcontrolador faz uso de uma placa Arduino conectada a um sensor infravermelho e foi adicionado com o objetivo de alterar a temperatura do ar-condicionado conforme comando recebido. Todos os dados coletados são enviados a um servidor local por meio de uma *ethernet shield*, acoplado a placa e conectado ao microcontrolador.

Figura 5.2 – Tomada Inteligente



Fonte: (SOLIGO, 2017).

Em resumo, os dados coletados pela tomada são: tensão, corrente consumida pelo equipamento, potência (calculada a partir da multiplicação da corrente com a tensão), temperatura, data e hora de cada coleta, a qual é realizada aproximadamente a cada 7 segundos. Estes dados podem ser considerados informações do contexto em nossa aplicação.

A temperatura adequada para o ambiente (valor ótimo) é definida pelo usuário deste. Porém, para demais valores como, por exemplo, valores adequados para consumo e variáveis necessárias ao controlador (como o deslocador u), sugere-se a aplicação do conhecimento de um especialista (ii), o qual é capaz de prover informações pertinentes ao domínio. Como este trabalho não obteve o conhecimento direto de um especialista, estes valores foram simulados.

Com base nisto, a Tabela 5.1 apresenta os intervalos finitos de possíveis valores de entrada (consumo e temperatura), valores de saída (configuração) e também os valores considerados ótimos para cada variável. Como pode ser observado na tabela, os valores de temperatura podem variar de 0 a 40 (°C), tendo 22°C como valor ótimo. Para o intervalo de valores de consumo (potência), o valor mínimo de entrada é 600 (W) e o máximo 2.500 (W). Como valor ótimo, foi definido o valor 1.500. A variável de saída do controlador corresponde a um valor de configuração a ser utilizado na alteração da temperatura do ar condicionado, ou seja, corresponde ao número de graus que serão aumentados ou redu-

zidos no ar condicionado. Estes valores variam de -7.5 a 7.5, além de um valor de alerta (AL) que ocorre em caso de valores de temperatura ou consumo extremamente altos ou baixos.

Tabela 5.1 – Valores mínimo, máximo e ótimo para variáveis de entrada e saída.

Variável	Intervalo	Valor ótimo
Temperatura (°C)	0 a 40	22
Consumo (W)	600 a 2500	1500
Configuração	-7.5 a 7.5 + AL	0

Fonte: Próprio autor.

A partir destes dados, a aplicação foi implementada seguindo o modelo definido neste trabalho.

5.1.1 Problema Multiobjetivo

Para o problema proposto, dois objetivos devem ser identificados:

obj_1 : (conforto térmico, {temperatura});

obj_2 : (eficiência energética, {potencia})

Com isto, o problema consiste em um conjunto contendo ambos objetivos:

P : { obj_1 , obj_2 }

De acordo com esta definição, a variável utilizada para obtenção do conforto térmico é 'temperatura' e para a eficiência energética 'potência'. No entanto, por conveniência, a partir de agora utilizaremos o termo 'consumo' para esta última.

5.1.2 Contexto de Interesse

No cenário descrito, as entidades físicas que definem o contexto do ambiente são: escritório, usuário, ar-condicionado, tomada inteligente, computador, sensores. As entidades abstratas, por sua vez, são: aplicação, problema multiobjetivo (P), dados de temperatura e consumo e serviço de modificação de temperatura. Considerando este cenário e os objetivos definidos para o problema, o contexto de interesse (Definição 2.2) foi representado a partir das seguintes relações semânticas:

```

    C = appHomeOffice{
<Escritorio, tem, ArCondicionadoX>
<Escritorio, tem, TomadaInteligenteX>
<Usuario, estaLocalizado, Escritorio>

<Objetivo1, temIntervalo, TemperaturaAdequada>
<Objetivo2, temIntervalo, ConsumoAdequado>
<P, temObjetivo, Objetivo1>
<P, temObjetivo, Objetivo2>

<ArCondicionadoX, estaLigadoA, TomadaInteligenteX>

<TomadaInteligenteX, temSensor, DHT11>
<TomadaInteligenteX, temSensor, SCT01320A>
<TomadaInteligenteX, temSensor, Tensao>
<TomadaInteligenteX, temSensor, InfravermelhoXX>

<TomadaInteligenteX, proveDado, ConsumoDeEnergiaX>
<TomadaInteligenteX, proveDado, TemperaturaX>

<TomadaInteligenteX, temServico, ModificaConfiguracaoArcondicio-
nado>
}

```

Ou seja, o contexto de interesse é representado pelo escritório que contém o ar condicionado, a tomada inteligente e um computador; pelo computador que executa a aplicação que processa o problema multiobjetivo (seção 5.1.1); pelo usuário que utiliza este ambiente; pelo ar condicionado conectado a tomada inteligente; pelos sensores existentes na tomada; pela leitura dos dados de temperatura e consumo de energia elétrica providos pela tomada inteligente e pelo serviço de alteração da configuração do ar condicionado oferecido pela tomada.

5.1.3 Situação de Interesse

De acordo com os objetivos definidos para este cenário, a situação de interesse é detectada quando os valores de consumo ou temperatura do ambiente não são considerados adequados. Portanto, com o objetivo de identificar a ocorrência de uma situação de interesse, porém evitando o processamento excessivo resultante da limitação dos valores definidos como ótimos, primeiramente foi utilizada lógica fuzzy para encontrar os intervalos

aceitáveis para as variáveis relacionadas aos objetivos. Para encontrar o conjunto de valores aproximados aos valores ótimos definidos na Tabela 5.1, foi utilizada a definição (2.7). Para cada variável, o valor de x corresponde aos valores contidos em seu intervalo finito, enquanto y representa o valor ótimo definido para a variável.

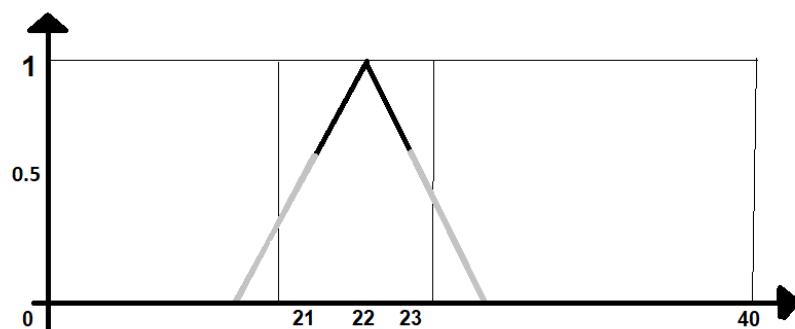
Considerando 0.7 como valor de grau de pertinência mínimo definido pelo especialista para a variável *temperatura*; 1.0 e 2.3 para os coeficientes a e b , respectivamente; e o valor ótimo definido na Tabela 5.1, o conjunto de valores equivalentes obtidos é representado na Tabela 5.2 e ilustrado na Figura 5.3. Estes valores correspondem ao conforto térmico do usuário.

Tabela 5.2 – Temperatura – Intervalo adequado para conforto térmico.

Valor	$\mu_{(\text{temperatura})}$
21	0.7
22	1.0
23	0.7

Fonte: Próprio autor.

Figura 5.3 – Intervalo adequado para temperatura



Fonte: Próprio autor.

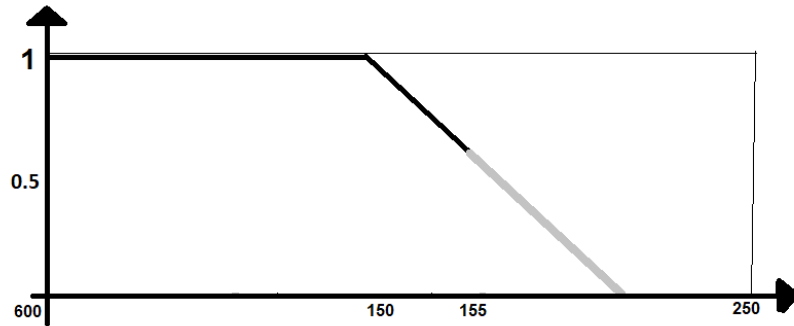
Tabela 5.3 – Consumo – Intervalo adequado para eficiência energética.

Valor	$\mu_{(\text{consumo})}$
150	1.0
151	0.93
152	0.88
153	0.82
154	0.76
155	0.7

Fonte: Próprio autor.

Para a variável *consumo*, por sua vez, foi utilizado o grau de pertinência mínimo 0.7 e 1.0 e 0.6 para os coeficientes a e b , respectivamente (Tabela 5.3, Figura 5.4). No entanto,

Figura 5.4 – Intervalo adequado para consumo



Fonte: Próprio autor.

como o valor de consumo abaixo do valor ótimo é considerado aceitável, o intervalo adequado foi calculado apenas para valores acima deste valor. Estes valores correspondem a valores adequados para obtenção de eficiência energética.

Ou seja, para valores de entrada dentro destes intervalos em ambos objetivos, não existe a necessidade do acionamento do módulo de decisão. Considerando que dados de sensores têm um nível de acurácia, ou seja, por vezes são falhos ou imprecisos, a aplicação de lógica fuzzy neste contexto proporciona resultados mais condizentes com a realidade.

Finalmente, de acordo com o modelo definido na seção 2.1, o conjunto de eventos que influenciam a situação de interesse é composto por apenas um evento (ev_1) identificando que um ou outro valor está inadequado (Quadro 5.1). A situação de interesse tem seu fim (Quadro 5.2) quando o oposto ocorrer, ou seja, quando os valores de consumo e temperatura forem considerados adequados.

Quadro 5.1 – Representação do evento ev_1 .

Nome do evento:	ev_1
Descrição:	valor de temperatura OU consumo é considerado inadequado
Tipo:	Externo
Tempo:	HoraMinSeg
{R}	$\langle \text{ConsumoDeEnergiaX}, \text{temValorColetado}, \text{TomadaInteligenteX} \rangle$ $\langle \text{Temperatura}, \text{temValorColetado}, \text{DHT11} \rangle$ $\Rightarrow \{ \text{ConsumoDeEnergiaX} > \text{optimalConsInterval} \vee$ $\text{Temperatura} \notin \text{optimalTempInterval} \}$
Padrão	$\text{optimalTempInterval} = \{21, 22, 23\},$ $\text{optimalConsInterval} = \{ \dots 150, 151, 152, 153, 154, 155 \}$

Fonte: Próprio autor.

Quadro 5.2 – Representação do evento ev_2 .

Nome do evento:	ev_2
Descrição:	valor de temperatura E consumo são considerados adequados
Tipo:	Externo
Tempo:	HoraMinSeg
{R}	$\langle \text{ConsumoDeEnergiaX}, \text{temValorColetado}, \text{TomadaInteligenteX} \rangle$ $\langle \text{Temperatura}, \text{temValorColetado}, \text{DHT11} \rangle$ $\Rightarrow \{ \text{ConsumoDeEnergiaX} \in \text{optimalConsInterval} \wedge \text{Temperatura} \in \text{optimalTempInterval} \}$
Padrão	$\text{optimalTempInterval} = \{21, 22, 23\},$ $\text{optimalConsInterval} = \{\dots 150, 151, 152, 153, 154, 155\}$

Fonte: Próprio autor.

Desta forma, a situação de interesse é definida como:

```

Si: (
nome: "Dados não satisfazem objetivos",
evSet: {ev1},
Fe: ev2)

```

5.1.4 Módulo de decisão

A partir da detecção da situação de interesse, são repassados os valores coletados pelos sensores e o processo de tomada de decisão é iniciado.

No desenvolvimento do controlador \mathcal{L} -fuzzy, para identificação das variáveis linguísticas a partir do conceito de valor 'ótimo' (Q_0), foram aplicados modificadores conforme as variáveis linguísticas (entrada e saída) representadas na Tabela 5.4.

Tabela 5.4 – Variáveis linguísticas de entrada e saída.

Temperatura	Consumo	Configuração
OT, //temperatura ótima	OC, //consumo ótimo	NC, //sem alteração
SW, //levemente quente	SH, //levemente alto	PS, //pequeno positivo
SC, //levemente frio	SL, //levemente baixo	NS, //pequeno negativo
LW, //pouco quente	LH, //pouco alto	PM, //médio positivo
LC, //pouco frio	LL, //pouco baixo	NM, //médio negativo
TW, //muito quente	TH, //muito alto	PB, //grande positivo
TC, //muito frio	TL, //muito baixo	NB, //grande negativo
MW, //extremamente quente	MH, //extremamente alto	AL; //alerta
MC; //extremamente frio	ML; //extremamente baixo	

Fonte: Próprio autor.

As relações \mathcal{L} -fuzzy correspondentes às variáveis linguísticas são dadas no Quadro

5.3:

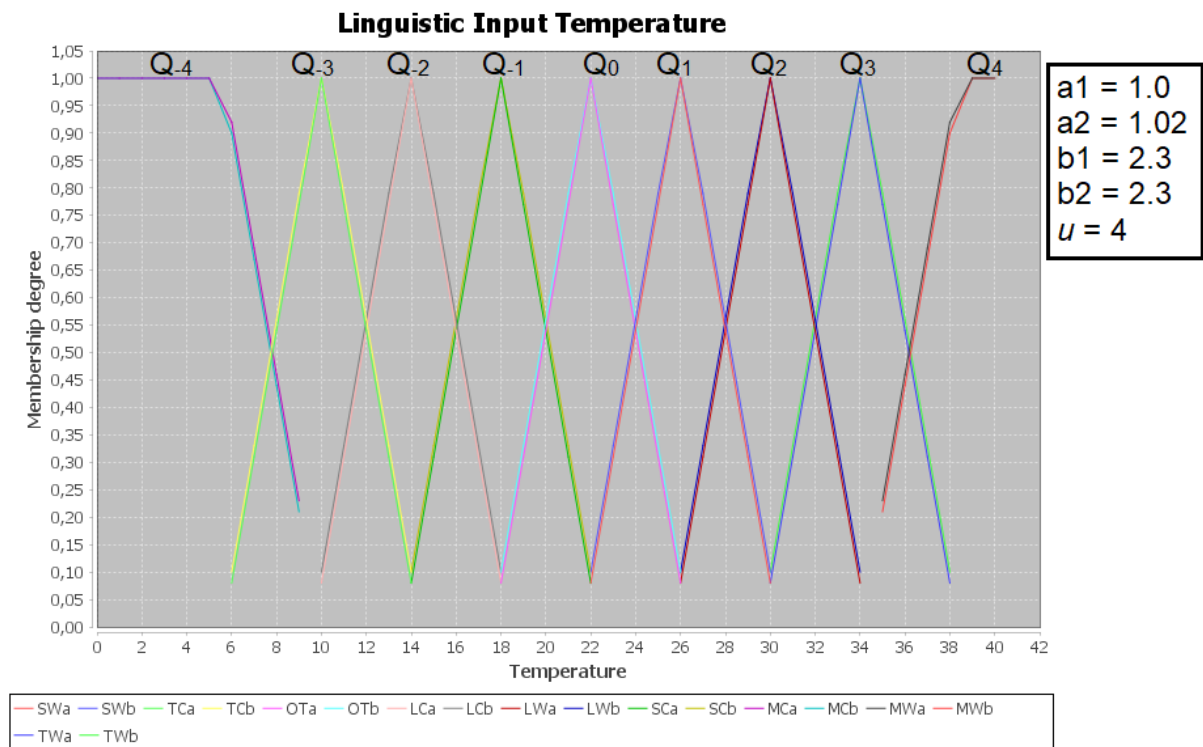
Quadro 5.3 – Relações \mathcal{L} -fuzzy correspondentes às variáveis linguísticas.

OT	Q ₀			OT	Q ₀			NC	S ₀	AL	S ₄
SW	Q ₁	SC	Q ₋₁	SH	Q ₁	SL	Q ₋₁	PS	S ₁	NS	S ₋₁
LW	Q ₂	LC	Q ₋₂	LH	Q ₂	LL	Q ₋₂	PM	S ₂	NM	S ₋₂
TW	Q ₃	TC	Q ₋₃	TH	Q ₃	TL	Q ₋₃	PB	S ₃	NB	S ₋₃
MW	Q ₄	MC	Q ₋₄	MH	Q ₄	ML	Q ₋₄				

Fonte: Próprio autor.

A partir dos valores ótimos identificados na Tabela 5.1, foram gerados os gráficos de temperatura (Gráfico 5.1), consumo (Gráfico 5.2) e de configuração (Gráfico 5.3).

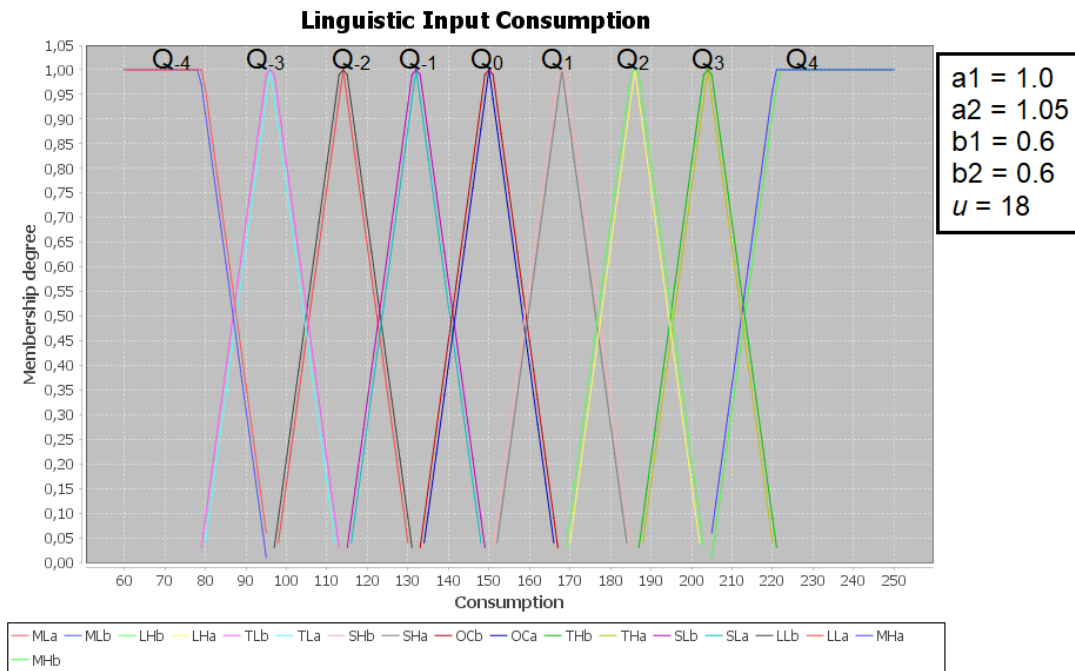
Gráfico 5.1 – Gráfico variável de entrada: Temperatura.



Fonte: Próprio autor.

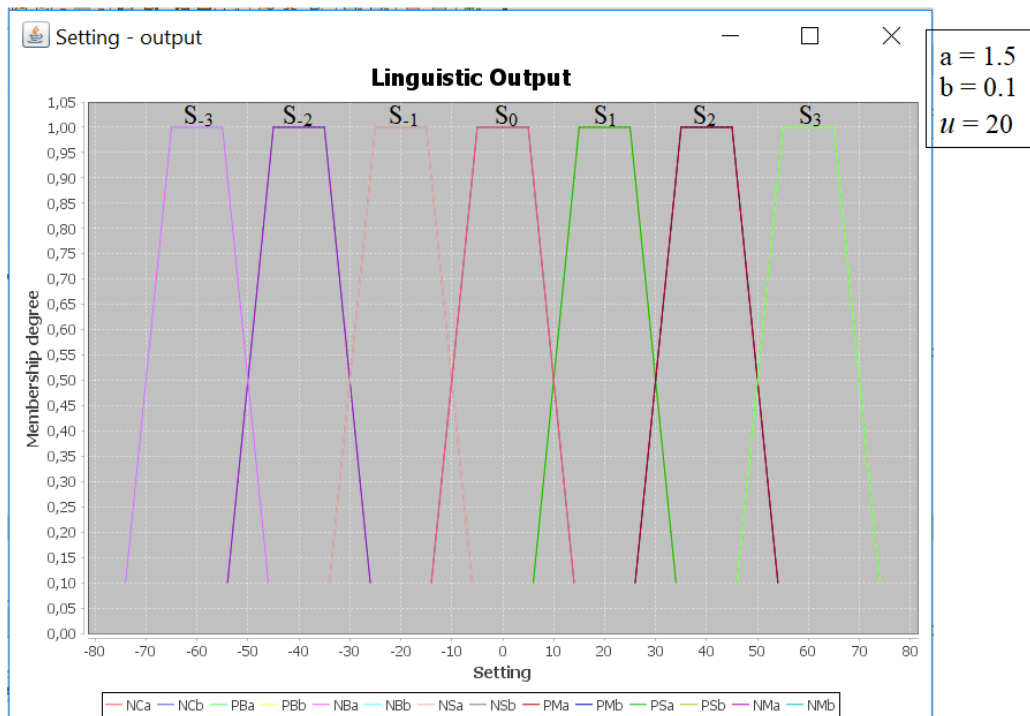
Com o objetivo de facilitar a aplicação das regras de inferência, os valores de entrada são reunidos em um *lattice* \mathcal{L} do tipo $[0,1] \times [0,1]$ por meio da aplicação da *t-norm de Łukasiewicz* (2.7). Desta forma, um novo conjunto (*L-rel*) de variáveis linguísticas é criado contendo pares de valores de temperatura e consumo. Neste trabalho, o método utilizado para fuzzificação destes valores foi a identidade do elemento.

Gráfico 5.2 – Gráfico variável de entrada: Consumo.



Fonte: Próprio autor.

Gráfico 5.3 – Gráfico variável de saída: Configuração.



Fonte: Próprio autor.

Com relação a base de regras, as regras de inferência utilizadas neste trabalho são representadas no Quadro 5.4.

Quadro 5.4 – Regras de inferência.

	OC	SH	SL	LH	LL	TH	TL	MH	ML
OT	NC	PS	NC	PS	NC	PS	NC	AL	NC
SW	NS	NC	NS	NS	NS	NC	NS	AL	NS
SC	PS	NC	PS	PS	PS	PS	PS	AL	PS
LW	NS	NS	NS	NC	NM	NC	NM	AL	NM
LC	PS	PS	PS	PS	PS	PS	PS	AL	PM
TW	NM	NS	NM	NS	NM	NC	NM	AL	NB
TC	PM	PM	PM	PM	PB	PM	PM	AL	PB
MW	NM	NS	NM	NS	NM	NS	NM	AL	NB
MC	PB	PB	PB	PB	PB	PB	PB	AL	PB

Fonte: Próprio autor.

Ou seja, para valor de consumo LL (*pouco baixo*) e temperatura LC (*pouco frio*) a regra aplicada será PS (*pequeno positivo*).

Neste trabalho, o processo de defuzzificação foi implementado considerando um valor ' α -cut' (0.3, 0.3), ou seja, espera-se um grau mínimo de 0.3 tanto para o primeiro objetivo quanto para o segundo. Estes são valores que devem ser definidos pelo especialista. No entanto, para fins de simulação neste trabalho, eles foram definidos durante testes de configuração do controlador.

Por questões de eficiência, como o conjunto resultante da inferência pode ser consideravelmente grande, não foram mapeados todos os valores para ínfimo ou supremo, como sugerido anteriormente. Ao invés disso, foi aplicada a operação α -cut no conjunto resultante e os valores iguais ou acima deste valor foram colocados em um novo conjunto, atingindo assim o mesmo resultado. Este novo conjunto contém todos os valores considerados aceitáveis para serem usados na configuração do ar condicionado. No entanto, um único valor deve ser selecionado como saída. Para seleção deste valor, o método de defuzzificação foi implementado considerando:

- se existir algum valor de alerta no conjunto, o resultado de saída *crisp* será 'ALERTA'
- se não existir alerta, o valor encontrado no meio do conjunto será o resultado final

A Tabela 5.5 mostra exemplos de dados de entrada, de um possível conjunto resultante e a saída do controlador:

Tabela 5.5 – Valores obtidos na saída do LFC de acordo com valores de entrada.

Entrada (temperatura, consumo)	Conjunto de Resultados	Saída <i>crisp</i> (configuração)
(23, 207)	-6.8,-6.7,...,-3.4,-2.0,...,7	-3.4
(5, 200)	2.0,...,3.8,...,5.2,...,7.0,AL	AL

Fonte: Próprio autor.

5.1.5 Ação automatizada

No cenário descrito, existe uma situação pré-condição para a execução da ação automatizada. Esta situação segue a definição de uma situação atual (2.4), a qual apresenta como evento de início o mesmo evento de início da situação de interesse (Quadro 5.1) e, como evento de fim, um evento identificando que uma decisão foi tomada pelo controlador \mathcal{L} -fuzzy (Quadro 5.5).

Quadro 5.5 – Representação do evento ev_3 .

Nome do evento:	ev_3
Descrição:	Controlador \mathcal{L} -fuzzy gerou saída <i>crisp</i>
Tipo:	Interno
Tempo:	HoraMinSeg
{R}	$\langle \text{LFC}, \text{temSaida}, \text{ValorConfiguracao} \rangle$ $\Rightarrow \{ \text{ValorConfiguracao} \in \text{IntervaloConfiguracao} \}$
Padrão	$\text{IntervaloConfiguracao} = \{-7, 5 \dots 7, 5, \text{AL}\}$

Fonte: Próprio autor.

Portanto, a situação pré-condição é definida como:

Sp_1 : (
 nome: "Decisão foi tomada",
 Ie: $\{ev_1\}$,
 Fe: $\{ev_3\}$)

Considerando que o resultado obtido pode ser um alerta (AL), primeiramente deve ser definido o que este alerta representa. Uma opção, seria disparar um alerta para o usuário decidir que ação deve ser executada. No entanto, com o objetivo de agir automaticamente e não necessitar intervenção do usuário, foi definido que este alerta deve ser convertido para um valor numérico. Para isso, deve ser verificado qual foi a última ação executada pelo sistema. Se esta ação "reduziu" a temperatura do ambiente, nada deve ser feito, pois o alerta é apenas uma consequência desta ação, a qual resultou no reacionamento do compressor do ar condicionado. No entanto, se "reduzir" não foi a última ação tomada, com o objetivo de reduzir o consumo atual, a temperatura do ar condicionado deve ser aumentada em 1 °C. Com isto objetiva-se a redução do esforço do ar condicionado (desligar o compressor).

Desta maneira, a ação automatizada pode ser representada da seguinte forma:

Nome: Altera configuração do ar condicionado

{Sp}: {Sp₁}

Serviços utilizados {Se}

Se: modificarConfiguracao;

{I}: TomadaInteligenteX, "ValorConfiguracao";

{O}: Serviço Interno, Servico_ajuste.

Evento resultante da ação {E}

Nome: Configuração alterada

Tipo: Interno

{R}: <TomadaInteligenteX, temServico, ModificaConfigArcondicionado>,

<ValorConfiguracao, executadoPor, TomadaInteligenteX>

Esta ação deve ser executada por meio do envio do valor *ValorConfiguracao* ao infra-vermelho acoplado a tomada inteligente. *ValorConfiguracao* é o valor obtido na tomada de decisão, o qual será somado ou subtraído à temperatura atualmente configurada no ar condicionado. Ou seja, se a temperatura configurada no ar condicionado for 21 e o valor de configuração for 3.0, o novo valor que o ar condicionado deve receber será 24.

No entanto, para os testes realizados neste trabalho, foi necessário executar esta ação manualmente utilizando o controle remoto do ar condicionado. Isto ocorreu devido a uma limitação de memória existente no microcontrolador, a qual impossibilitou a clonagem de todos os valores que podem ser configurados no ar condicionado. Porém, considerando que a temperatura será de fato alterada sempre que definido pelo sistema, esta limitação não afeta os resultados dos testes, sendo inconveniente apenas em uma situação de utilização real do sistema

6 RESULTADOS

Para viabilizar o ambiente descrito no capítulo anterior, foi utilizada uma sala de aulas do colégio Politécnico da UFSM¹, sendo que os testes foram realizados em períodos em que a sala estava vaga e, portanto, não apresentava entrada e saída de pessoas. Durante a realização destes testes, os únicos equipamentos eletrônicos utilizados foram dois computadores, os quais afetam minimamente a temperatura da sala. Visando não afetar os resultados, aberturas da porta de entrada do ambiente de teste ocorreram em torno de apenas 4 vezes para cada dia de teste, as quais eram feitas rapidamente apenas para permitir a entrada e saída do usuário da sala.

Com o objetivo de verificar a eficácia do sistema desenvolvido, foram realizados testes neste ambiente através de três cenários: a) utilizando a aplicação, b) não utilizando esta e c) utilizando a aplicação, porém sem detectar a situação de interesse. O terceiro cenário teve o objetivo de identificar a contribuição da detecção da situação de interesse para o sistema. Como descrito neste trabalho, a aplicação visa monitorar o ambiente por meio da tomada de decisões que mantenham o conforto térmico ao mesmo tempo que se preocupam com a eficiência energética. Sem o uso da aplicação este monitoramento não existe e, conseqüentemente, é realizado apenas o controle da temperatura do ambiente sendo deixada de lado a eficiência energética. Este controle é feito por meio da configuração atribuída ao ar condicionado quando ele é ligado ou quando esta é alterada pelo controle remoto.

Como os testes foram realizados utilizando o mesmo ambiente, eles foram realizados em dias diferentes, no entanto, buscamos avaliar dados na mesma faixa horária e em condições climáticas semelhantes (Tabela 6.1). Isto justifica a falta de sequência na ordem de execução destes nos três cenários, como pode ser visto na Tabela 6.2. Os testes foram realizados entre 02 e 19 de janeiro de 2018, somando um total de 82 horas, das quais 31 utilizaram a aplicação, 31 não utilizaram e 20 utilizaram a aplicação, porém sem a detecção da situação de interesse.

Tabela 6.1 – Temperatura média (externa) por tipo de teste. a) utilizando aplicação, b) não utilizando aplicação, c) utilizando aplicação sem detectar a situação de interesse (Si)

Tipo	Temperatura Média (°C)
a	28.8
b	29.5
c	27.8

Fonte: Próprio autor.

O arquivo contendo todos os registros obtidos nos testes pode ser encontrado em

¹<http://www.politecnico.ufsm.br/>

Tabela 6.2 – Dados da execução dos testes. a) utilizando aplicação, b) não utilizando aplicação, c) utilizando aplicação sem Si

Data	Início	Fim	Total Horas	Temp Min	Temp Máx	Tipo
02/01/2018	10:27	17:33	7	23	28	a
03/01/2018	11:11	19:15	6	28	31	a
04/01/2018	11:14	18:25	7	30	34	a
05/01/2018	10:16	12:16	2	29	31	a
09/01/2018	11:06	18:06	7	23	31	b
10/01/2018	10:52	17:32	6	28	34	b
11/01/2018	11:56	18:56	7	28	35	b
12/01/2018	12:23	18:26	6	28	33	c
15/01/2018	09:30	18:43	9	23	29	a
16/01/2018	09:33	17:33	8	22	30	c
17/01/2018	12:24	14:24	2	27	30	b
18/01/2018	09:32	18:33	9	25	30	b
19/01/2018	09:26	15:51	6	23	30	c

Fonte: Próprio autor.

<https://github.com/milenetsi/moami/blob/master/testes.xlsx>.

A aplicação foi configurada para executar seu raciocínio a cada 10 minutos, gerando 193 registros para o total 31 horas de execução. Foram identificadas 110 ocorrências da situação de interesse (57%), a qual foi respeitada não ativando o módulo de decisão quando a temperatura e consumo estavam dentro dos intervalos tolerados. Nestas ocorrências, a temperatura foi reduzida 36 vezes, aumentada 49 e mantida 25 vezes, conforme decisão tomada pela aplicação. A Figura 6.1 mostra alguns registros obtidos, incluindo a data e hora, temperatura e consumo utilizados como entrada no sistema, temperatura externa ao ambiente, situação de interesse ativa ou não, saída recebida do controlador, valor aplicado no ar condicionado e ação executada.

Figura 6.1 – Amostra de registros utilizando a aplicação.

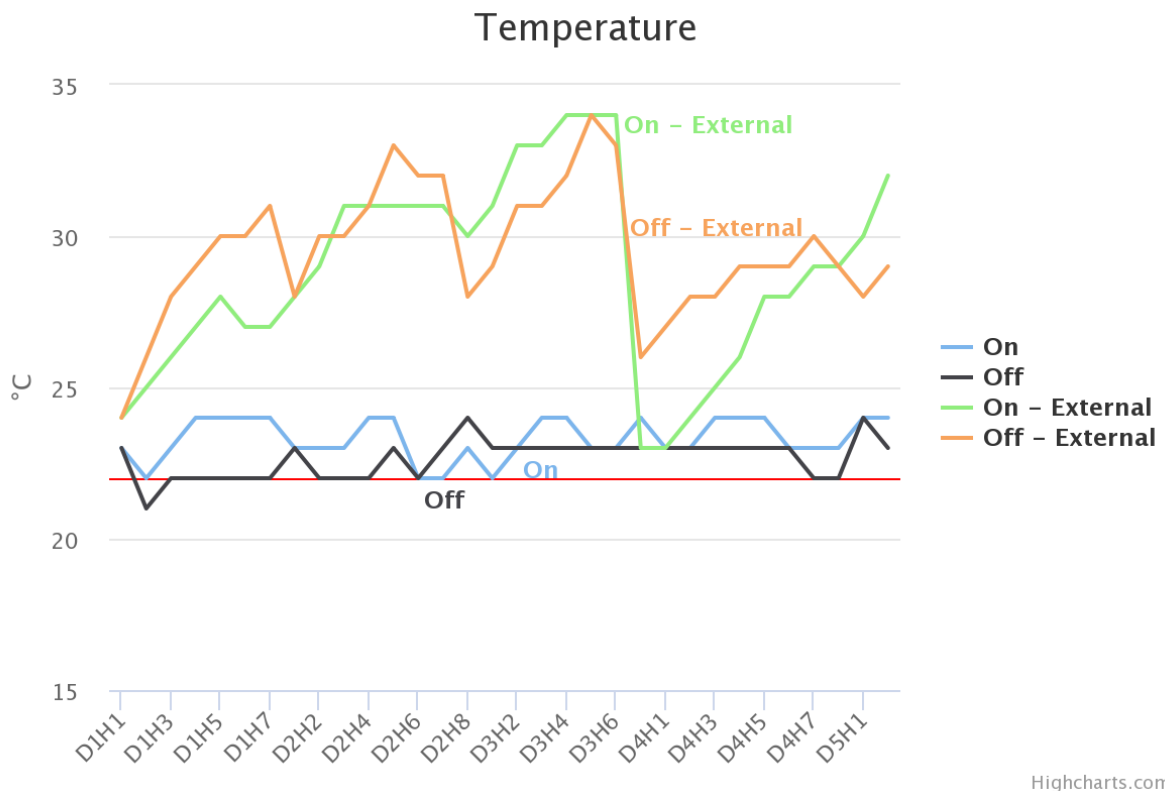
	Temperature	Consumption	Ext. Temp	Situation	On	Orig setting	Setting	Action
15/01/2018 16:13	23	63	28	0	-	-	-	-
15/01/2018 16:23	24	194	28	1	-3,1	-3	Reduce	
15/01/2018 16:33	23	224	28	1	AL	1	Increase	
15/01/2018 16:43	23	222	29	1	AL	1	Increase	
15/01/2018 16:53	23	117	29	0	-	-	-	
15/01/2018 17:03	23	121	29	0	-	-	-	
15/01/2018 17:13	23	157	28	1	-0,2	0	Keep	
15/01/2018 17:23	23	164	28	1	1,9	2	Increase	
15/01/2018 17:33	23	96	28	0	-	-	-	
15/01/2018 17:43	24	60	29	1	-3	-3	Reduce	
15/01/2018 17:53	24	148	29	1	0,6	1	Increase	
15/01/2018 18:03	23	177	28	1	1	1	Increase	
15/01/2018 18:13	23	107	28	0	-	-	-	
15/01/2018 18:23	23	96	28	0	-	-	-	
15/01/2018 18:33	23	86	29	0	-	-	-	

Fonte: Próprio autor.

Os testes sem o uso da aplicação, os quais também somaram 31 horas, foram realizados utilizando o mesmo ambiente físico. Porém, neste cenário, o ar condicionado foi configurado em 22°C (temperatura ótima, conforme tabela 5.1), não sofrendo alterações durante o período dos testes. Para este cenário também foram registrados valores correspondentes a temperatura interna do ambiente, consumo gerado e temperatura externa.

O Gráfico 6.1 apresenta os valores de média horária de temperatura interna e externa obtidos nos cenários *a* e *b*. Apenas olhando o gráfico, os valores que representam a utilização da aplicação aparentam ser menos adequados que os valores sem o uso desta. No entanto, isso ocorreu em consequência do controle do consumo energético, pois com o objetivo de controlar o consumo, a temperatura é afetada e vice versa. Contudo, apesar de apresentar certa diferença, ambos cenários mantiveram o conforto térmico do ambiente na maior parte do tempo.

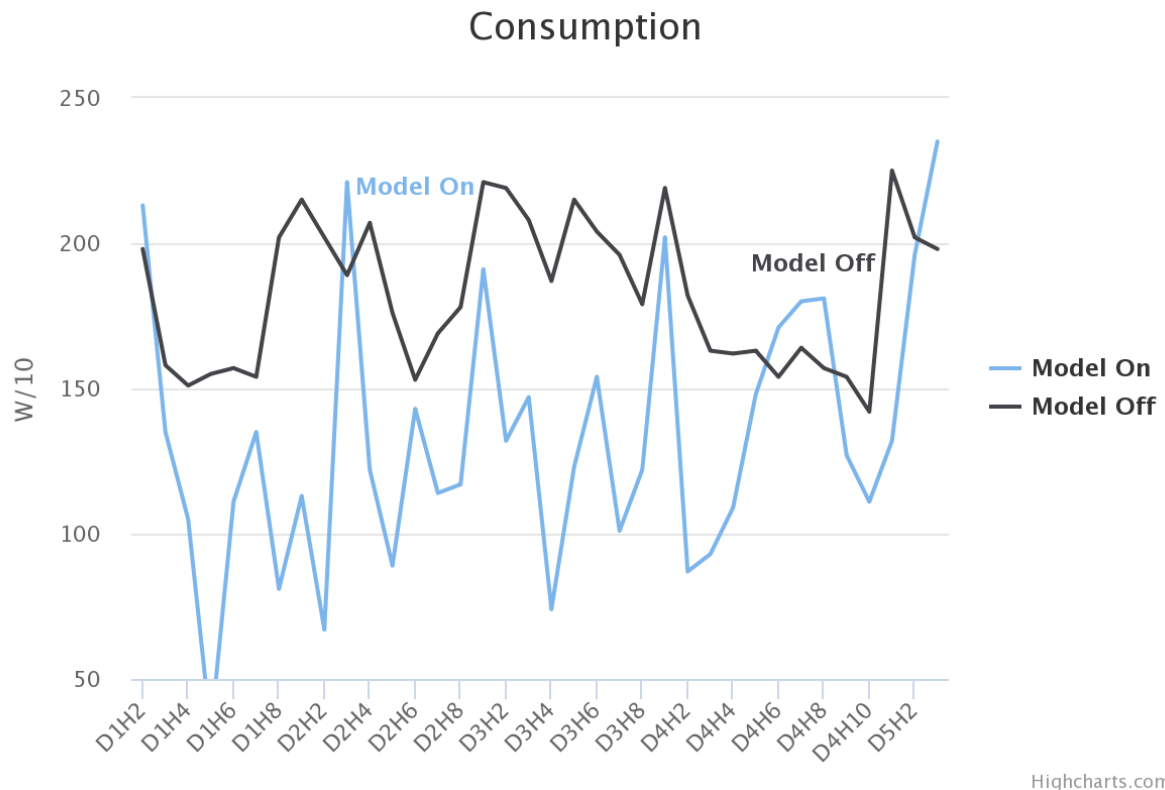
Gráfico 6.1 – Temperatura interna e externa com e sem uso da aplicação. No eixo x, D: dia, H:hora



Fonte: Próprio autor.

Com relação ao consumo energético, pode ser notada uma diferença maior nos resultados (Gráfico 6.2). A utilização da aplicação possibilitou um consumo relativamente inferior ao consumo normal, havendo poucas ocorrências de valores superiores. Isto foi possível devido aos reajustes de temperatura feitos pela aplicação, os quais permitiam o aumento desta em situações em que ela estava adequada, mas o consumo estava alto.

Gráfico 6.2 – Consumo com e sem uso da aplicação. No eixo x, D: dia, H:hora



Fonte: Próprio autor.

Em resumo, de acordo com os dados obtidos para as 31 horas utilizando a aplicação, a temperatura média da sala foi de 23°C e o consumo 1.337W. Para os dados obtidos sem o uso da aplicação, considerando o intervalo de 10 minutos, também foi obtida uma temperatura média da sala de 23°C, porém o consumo foi de 1.827W. A partir destes dados, podemos identificar que a temperatura do ambiente é mantida 'agradável' (Tabela 5.2) ao usuário em ambos casos, porém o uso da aplicação permitiu uma economia de 27% no consumo de energia.

Por fim, os testes realizados sem a detecção da situação de interesse apresentaram resultados similares aos que detectam esta. No entanto, a temperatura foi reajustada a cada iteração (10 minutos), o que causa uma demanda desnecessária na execução da ação automatizada. A Figura 6.2 exibe uma amostra dos dados obtidos neste cenário. Como pode ser observado, as colunas que identificam os valores de entrada utilizados para temperatura e consumo, frequentemente apresentam valores que seriam considerados adequados (em vermelho), não havendo necessidade de uma tomada de decisão. Em consequência, como saída foram obtidos valores baixos solicitando a redução, aumento ou então para manter a temperatura. Como resultado, dos 125 registros obtidos neste cenário, 41 são "falsos positivos", o que representa 33% de ações automatizadas executadas desnecessariamente. Como médias, foram obtidos os valores de 23°C e 1.300W para

Figura 6.2 – Amostra de registros utilizando a aplicação, porém sem Situação de Interesse (Si).

Datetime	Temperature	Consumption	Ext. Temp	Orig setting	Setting	Action
19/01/2018 09:26	22	177	23	1	1	Increase
19/01/2018 09:36	22	196	23	1,6	2	Increase
19/01/2018 09:46	22	103	23	0	0	Keep
19/01/2018 09:56	22	60	23	-1,1	-1	Reduce
19/01/2018 10:06	23	102	23	-0,9	-1	Reduce
19/01/2018 10:16	22	139	25	1	1	Increase
19/01/2018 10:26	22	84	25	-1,1	-1	Reduce
19/01/2018 10:36	22	132	25	-0,5	0	Keep
19/01/2018 10:46	22	131	25	-1,1	-1	Reduce
19/01/2018 10:56	22	164	25	0,5	1	Increase
19/01/2018 11:06	22	155	25	0,7	1	Increase
19/01/2018 11:16	22	68	25	1	1	Increase
19/01/2018 11:26	22	60	27	1	1	Increase
19/01/2018 11:36	23	60	27	0,4	0	Keep
19/01/2018 11:46	23	60	27	-0,4	0	Keep
19/01/2018 11:56	24	60	27	-1,2	-1	Reduce

Fonte: Próprio autor.

temperatura e consumo, respectivamente.

A Tabela 6.3 resume os resultados para os três cenários.

Tabela 6.3 – Resultados dos testes: valores médios conforme cenário

Tipo	Temperatura (°C)	Consumo (W)
Com app	23	1.337
Sem app	23	1.827
Com app, sem Si	23	1.300

Fonte: Próprio autor.

6.1 CONSIDERAÇÕES

Com o uso desta aplicação, o usuário não precisa se preocupar em alterar a temperatura tendo o receio de estar consumindo muita energia. Utilizando o conhecimento do especialista, a aplicação fica encarregada deste raciocínio, tomando decisões e agindo apenas quando e conforme o necessário. Isto traz inteligência para o ambiente, possibilitando que o usuário foque em suas atividades.

A detecção da situação de interesse contribui para evitar a demanda excessiva de alterações na configuração do ar condicionado. Isto ocorre, pois ao invés de ter que

executar uma ação a cada e toda iteração do sistema, o sistema precisa prestar atenção apenas nas situações de interesse, ou seja, que apresentam valores inadequados.

Os valores obtidos nos cenários de teste não são precisos. Alguns fatores podem afetar os resultados obtidos como, por exemplo, a temperatura externa do ambiente. Para a realização dos testes, não foi possível obter dias e horários com os mesmos exatos valores de temperatura externa, portanto este fator pode ter influenciado o resultado final. Outro fator que pode influenciar o resultado é o conhecimento do especialista aplicado. Para fins de testes, nesta aplicação este conhecimento foi simulado. Acreditamos que com a real utilização do conhecimento de um especialista, os resultados obtidos poderiam ser ainda mais satisfatórios. Apesar disso, consideramos que, mesmo que possa apresentar um índice de erro, 27% é um valor significativo para economia de energia. Desta forma, a utilização deste sistema é validada e justificada em um ambiente inteligente.

7 CONCLUSÃO

O presente trabalho teve como objetivo desenvolver uma estratégia para tomada de decisão multiobjetiva em Ambientes Inteligentes (Aml) para problemas com objetivos de mesma importância. Esta estratégia consiste na definição de um modelo sensível ao contexto que, a partir da detecção de uma situação de interesse, permite aos sistemas agirem automaticamente de acordo com uma decisão que é tomada utilizando a teoria \mathcal{L} -fuzzy. Este modelo pode ser utilizado no desenvolvimento de sistemas para Aml que buscam reproduzir o raciocínio humano na solução de problemas multiobjetivos.

O modelo foi desenvolvido utilizando modelos e teorias definidos nos trabalhos de (MACHADO et al., 2017) e (WINTER, 2007). Para sua definição, primeiramente foi realizado um estudo sobre os conceitos de: sensibilidade ao contexto e à situação, tomada de decisão multicritérios, lógica fuzzy e sua extensão \mathcal{L} -fuzzy e ações automatizadas.

Posteriormente, foram apresentados trabalhos encontrados na literatura que possuem semelhanças aos diferentes tópicos aplicados na definição do modelo proposto. Como não foi encontrado nenhum trabalho que aborde todos os conceitos aplicados, foram buscados trabalhos que apresentem maior semelhança com os diferentes módulos do modelo. Os trabalhos encontrados que abordam a tomada de decisão multiobjetiva, fazem uso de pesos aos critérios, os quais afetam o resultado final. Este trabalho visa a não priorização dos critérios, ou seja, considera que nenhum parâmetro pode ser considerado mais ou menos importante que o outro, portanto não utiliza pesos.

Para verificação do modelo, como exemplo de ambiente com um problema multiobjetivo, foi utilizado o cenário do escritório de um programador. Para este ambiente, utilizando o modelo proposto, foi desenvolvido um sistema que visa manter o conforto térmico ao mesmo tempo que busca otimizar a eficiência energética. Foram realizados testes comparando este cenário a dois outros: a) sem a utilização do sistema e b) utilizando o sistema, mas sem detectar a situação de interesse.

Os resultados obtidos permitiram identificar que a não detecção da situação de interesse gera uma demanda excessiva do sistema, agindo desnecessariamente quando a situação estava adequada. Em contrapartida, a utilização do sistema detectando a situação de interesse, possibilitou o alcance do conforto térmico na maior parte do tempo e uma economia de energia de 27%, quando comparado ao cenário sem a utilização do sistema. Isto responde positivamente a questão de pesquisa apresentada junto aos objetivos deste trabalho, pois o uso do sistema permitiu a manipulação da situação de interesse (que era uma situação indesejada).

A principal contribuição deste trabalho foi a disponibilização do modelo desenvolvido, o qual foi validado com o estudo de caso, podendo ser aplicado em outros domínios. Este modelo permite o raciocínio sobre a incerteza, considerando aspectos complexos e

incertos como o raciocínio humano na tomada de decisão. Além disso, a ocorrência de uma situação de interesse também é incerta e, por conta disto, o modelo sugere a utilização de ferramentas que facilitem a detecção desta, como foi o exemplo da utilização da lógica fuzzy para identificar os intervalos adequados no estudo de caso.

Além do modelo, este trabalho tem como contribuição a disponibilização da biblioteca \mathcal{L} -fuzzy, que pode ser utilizada para desenvolvimento de controladores para tomada de decisão multiobjetiva. Se desejado, esta biblioteca também pode ser utilizada para solução problemas com objetivo único, porém, como este não é o objetivo dela, sugere-se o uso de ferramentas direcionadas a resolver tais problemas. De mesma forma, ela pode ser utilizada com valores lineares, pois estes estão contidos nos valores não-lineares. No entanto, quando o problema em questão exigir resultados precisos, a tomada de decisão não deve ser feita com a utilização de lógica fuzzy. Nestes casos, ferramentas para raciocínio preciso devem ser utilizadas.

Como oportunidade para trabalhos futuros, sugere-se a adaptação deste modelo visando contemplar o aspecto proatividade com o intuito de antecipar ações necessárias no ambiente. Com o objetivo de atualizar automaticamente situações de interesse e o conhecimento utilizado para tomada de decisão, outra oportunidade para trabalhos futuros é a inclusão de características de sistemas de aprendizagem no modelo.

Uma primeira versão deste trabalho foi publicada no evento ICEIS: *International Conference on Enterprise Information Systems*, Portugal, Abril 2017, com o título “Fuzzy Based Model to Detect Patient’s Health Decline in Ambient Assisted Living” (Qualis B2).

Um segundo artigo está sendo finalizado e será submetido para o *Journal Sensors* - ISSN 1424-8220. (Qualis A1).

REFERÊNCIAS BIBLIOGRÁFICAS

- AARTS, E.; WICHERT, R. Ambient intelligence. In: **Technology Guide**. [S.l.]: Springer, 2009. p. 244–249.
- ABOWD, G. et al. Towards a better understanding of context and context-awareness. In: SPRINGER. **Handheld and ubiquitous computing**. [S.l.], 1999. p. 304–307.
- ATMEL, C. L. D.; BOOK, A. **Atmel Corporation, 1995**. [S.l.]: through.
- AUGUSTO, J. C.; MCCULLAGH, P. Ambient intelligence: Concepts and applications. **Computer Science and Information Systems**, v. 4, n. 1, p. 1–27, 2007.
- BAI, Y.; WANG, D. Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications. In: **Advanced Fuzzy Logic Technologies in Industrial Applications**. [S.l.]: Springer, 2006. p. 17–36.
- BARR, M.; WELLS, C. **Category theory for computing science**. [S.l.]: Prentice Hall New York, 1990. v. 1.
- BELLMAN, R. E.; ZADEH, L. A. Decision-making in a fuzzy environment. **Management science**, INFORMS, v. 17, n. 4, p. B–141, 1970.
- BIRKHOFF, G. **Lattice theory**. [S.l.]: American Mathematical Soc., 1940. v. 25.
- BOUSDEKIS, A. et al. A probabilistic model for context-aware proactive decision making. In: IEEE. **Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on**. [S.l.], 2016. p. 1–6.
- CHEN, S.-M.; TAN, J.-M. Handling multicriteria fuzzy decision-making problems based on vague set theory. **Fuzzy sets and systems**, Elsevier, v. 67, n. 2, p. 163–172, 1994.
- CORONATO, A.; PIETRO, G. D. Formal design of ambient intelligence applications. **Computer**, IEEE, v. 43, n. 12, p. 60–68, 2010.
- DANIELLO, G.; LOIA, V.; ORCIUOLI, F. Employing fuzzy consensus for assessing reliability of sensor data in situation awareness frameworks. In: IEEE. **Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on**. [S.l.], 2015. p. 2591–2596.
- DEB, K. Multi-objective optimisation using evolutionary algorithms: an introduction. In: **Multi-objective evolutionary optimisation for product design and manufacturing**. [S.l.]: Springer, 2011. p. 3–34.
- DERNONCOURT, F. Introduction to fuzzy logic. **Massachusetts Institute of Technology**, 2013.
- EL-WAHED, W. F. A. Intelligent fuzzy multi-criteria decision making: review and analysis. **Fuzzy Multi-Criteria Decision Making**, Springer, p. 19–50, 2008.
- ENDSLEY, M. Toward a theory of situation awareness in dynamic systems. **Human Factors: The Journal of the Human Factors and Ergonomics Society**, v. 1, n. 37, p. 32–64, 1995.

_____. **Designing for situation awareness: An approach to user-centered design.** [S.l.]: CRC Press, 2011.

ENDSLEY, M.; GARLAND, D. e. **Situation awareness analysis and measurement.** [S.l.]: CRC Press, 2000.

ENDSLEY, M.; JONES, W. *The oxford handbook of cognitive engineering.* 2013.

FANDEL, G.; GÁL, T.; HANNE, T. **Multiple criteria decision making.** [S.l.]: Springer, 1997.

FELSTEAD, A. et al. Opportunities to work at home in the context of work-life balance. **Human resource management journal**, Wiley Online Library, v. 12, n. 1, p. 54–76, 2002.

FENG, Y.-H.; TENG, T.-H.; TAN, A.-H. Modelling situation awareness for context-aware decision support. **Expert Systems with Applications**, Elsevier, v. 36, n. 1, p. 455–463, 2009.

FOWKES, L.; SANDERS, W. et al. **A survey of Remote Jobs and Communities Program (me) providers: one year in.** [S.l.]: Centre for Aboriginal Economic Policy Research, ANU, 2015.

GOGUEN, J. A. L-fuzzy sets. **Journal of mathematical analysis and applications**, Elsevier, v. 18, n. 1, p. 145–174, 1967.

GOVINDAN, K.; KHODAVERDI, R.; JAFARIAN, A. A fuzzy multi criteria approach for measuring sustainability performance of a supplier based on triple bottom line approach. **Journal of Cleaner Production**, Elsevier, v. 47, p. 345–354, 2013.

GRÄTZER, G. **General lattice theory.** [S.l.]: Springer Science & Business Media, 2002.

HILL, E. J.; FERRIS, M.; MÄRTINSON, V. Does it matter where you work? a comparison of how three work venues (traditional office, virtual office, and home office) influence aspects of work and personal/family life. **Journal of Vocational Behavior**, Elsevier, v. 63, n. 2, p. 220–241, 2003.

HONG, D. H.; CHOI, C.-H. Multicriteria fuzzy decision-making problems based on vague set theory. **Fuzzy sets and systems**, Elsevier, v. 114, n. 1, p. 103–113, 2000.

JOHNS, T.; GRATTON, L. The third wave of virtual work. **Harvard Business Review**, v. 91, n. 1, p. 66–73, 2013.

JONES, R. E. et al. Modeling situation awareness for army infantry platoon leaders using fuzzy cognitive mapping techniques. In: **Proceedings of the Behavior Representation in Modeling and Simulation (BRIMS) Conference.** [S.l.: s.n.], 2010. p. 216–223.

KAHRAMAN, C. **Fuzzy multi-criteria decision making: theory and applications with recent developments.** [S.l.]: Springer Science & Business Media, 2008. v. 16.

KLIR, G.; YUAN, B. **Fuzzy sets and fuzzy logic.** [S.l.]: Prentice hall New Jersey, 1995. v. 4.

KOKAR, M. M.; MATHEUS, C. J.; BACLAWSKI, K. Ontology-based situation awareness. **Information fusion**, Elsevier, v. 10, n. 1, p. 83–98, 2009.

LEEKWIJCK, W. V.; KERRE, E. E. Defuzzification: criteria and classification. **Fuzzy sets and systems**, Elsevier, v. 108, n. 2, p. 159–178, 1999.

MACHADO, A. Sensibilidade à situação em ambientes de vivência assistida: uma abordagem reativa, proativa e extensível. 2015.

MACHADO, A. et al. Reactive, proactive, and extensible situation-awareness in ambient assisted living. **Expert Systems with Applications**, Elsevier, v. 76, p. 21–35, 2017.

MAIO, C. D. et al. Swarm-based semantic fuzzy reasoning for situation awareness computing. In: IEEE. **Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on**. [S.l.], 2012. p. 1–7.

MARRO, A. A. et al. Lógica fuzzy: conceitos e aplicações. **Natal: Universidade Federal do Rio Grande do Norte (UFRN)**, 2010.

MIETTINEN, K. **Nonlinear multiobjective optimization**. [S.l.]: Springer Science & Business Media, 2012. v. 12.

MOHANTY, R. et al. A fuzzy anp-based approach to r&d project selection: a case study. **International Journal of Production Research**, Taylor & Francis, v. 43, n. 24, p. 5199–5216, 2005.

MUNARETTO, A. C. C. Da teoria de reticulados à geometria algébrica fuzzy. 2005.

NADERPOUR, M.; LU, J. A fuzzy dual expert system for managing situation awareness in a safety supervisory system. In: IEEE. **Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on**. [S.l.], 2012. p. 1–7.

NGATCHOU, P.; ZAREI, A.; EL-SHARKAWI, A. Pareto multi objective optimization. In: IEEE. **Intelligent systems application to power systems, 2005. Proceedings of the 13th international conference on**. [S.l.], 2005. p. 84–91.

PANTELI, M. et al. Assessing the impact of insufficient situation awareness on power system operation. **IEEE Transactions on power systems**, IEEE, v. 28, n. 3, p. 2967–2977, 2013.

PERNAS, A. M. et al. A semantic approach for learning situation detection. In: IEEE. **Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on**. [S.l.], 2014. p. 1119–1126.

RAMOS, C.; AUGUSTO, J. C.; SHAPIRO, D. Ambient intelligence—the next step for artificial intelligence. **IEEE Intelligent Systems**, IEEE, v. 23, n. 2, p. 15–18, 2008.

ROJAS, R. **Neural networks: a systematic introduction**. [S.l.]: Springer Science & Business Media, 2013.

SANTOS, J. E. S. D. et al. Controle preditivo não-linear para sistemas de hammerstein. Florianópolis, SC, 2007.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: IEEE. **Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on**. [S.l.], 1994. p. 85–90.

SHAIKH, P. H. et al. Intelligent multi-objective optimization for building energy and comfort management. **Journal of King Saud University-Engineering Sciences**, Elsevier, 2016.

SOLIGO, D. **INTERNET DAS COISAS APLICADA À EFICIÊNCIA ENERGÉTICA**. 2017.

SORGATO, M. J.; ROBERTO, L. **Relatório para especificação técnica de Condicionadores de Ar com eficiência energética “A”**. [S.l.], 2014. 13 p.

UHLARIK, J.; COMERFORD, D. **A review of situation awareness literature relevant to pilot surveillance functions**. [S.l.]: DIANE Publishing, 2002.

VIEIRA, R. d. S. et al. Protótipo de um sistema de monitoramento remoto inteligente. Florianópolis, SC, 1999.

WANG, Z.; YANG, R.; WANG, L. Multi-agent control system with intelligent optimization for smart and energy-efficient buildings. In: IEEE. **IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society**. [S.l.], 2010. p. 1144–1149.

WEBER, R. H.; WEBER, R. **Internet of things**. [S.l.]: Springer, 2010. v. 12.

WINTER, M. **Goguen categories: a categorical approach to L-fuzzy relations**. [S.l.]: Springer Science & Business Media, 2007. v. 25.

YAGER, R. R.; ZADEH, L. A. **An introduction to fuzzy logic applications in intelligent systems**. [S.l.]: Springer Science & Business Media, 2012. v. 165.

YANG, R.; WANG, L. Multi-objective optimization for decision-making of energy and comfort management in building automation and control. **Sustainable Cities and Society**, Elsevier, v. 2, n. 1, p. 1–7, 2012.

YU, W. et al. Application of multi-objective genetic algorithm to optimize energy efficiency and thermal comfort in building design. **Energy and Buildings**, Elsevier, v. 88, p. 135–143, 2015.

ZADEH, L. Z. Fuzzy sets. **Inf. Control**, v. 8, p. 338–353, 1965.