



Cesar Augusto Sierra Franco

**Técnicas de aprendizagem para gerência de recursos em
redes móveis heterogêneas e auto-organizáveis**

TESE DE DOUTORADO

Tese apresentada como requisito parcial para
obtenção do grau de Doutor pelo programa de Pós-
graduação em Engenharia Elétrica da PUC-Rio.

Orientador: Prof. Gláucio Lima Siqueira

Co-orientador: Prof. José Roberto Boisson de Marca

Rio de Janeiro
Abril de 2018



Cesar Augusto Sierra Franco

**Técnicas de aprendizagem para gerência de recursos em
redes móveis heterogêneas e auto-organizáveis**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-Graduação em Engenharia Elétrica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Gláucio Lima Siqueira

Orientador

Centro de Estudos e Telecomunicações – PUC-Rio

Prof. José Roberto Boisson de Marca

Co-Orientador

Centro de Estudos e Telecomunicações – PUC-Rio

Prof. Luiz Alencar Reis da Silva Mello

Centro de Estudos e Telecomunicações – PUC-Rio

Prof. Andre Noll Barreto

UNB

Prof. Paulo Cardieri

UNICAMP

Prof. Moises Vidal Ribeiro

UFJF

Prof. Márcio da Silveira Carvalho

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 09 de abril de 2018

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, da autora e do orientador.

Cesar Augusto Sierra Franco

Engenheiro em Eletrônica e Telecomunicações pela Universidad del Cauca - Colômbia (2009) e Mestre em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2013.

Ficha Catalográfica

Sierra Franco, Cesar Augusto

Técnicas de aprendizagem para gerência de recursos em redes móveis heterogêneas e auto-organizáveis / Cesar Augusto Sierra Franco; orientador: Gláucio Lima Siqueira; co-orientador: José Roberto Boisson de Marca. – 2018.

134 f. : il. ; 30 cm

Tese (doutorado)—Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2018.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Redes móveis auto-organizáveis. 3. Redes heterogêneas. 4. Redes cognitivas. 5. Balanceamento de carga. I. Siqueira, Gláucio Lima. II. Boisson de Marca, José Roberto. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

À Greis, minha companheira de vida, sonhos e ilusões.

Agradecimentos

A Deus, por esta maravilhosa oportunidade e por ter colocado no meu caminho tudo quanto foi preciso para culminar esta etapa.

À Greis, o amor da minha vida, que fez este sonho possível e sempre esteve ao meu lado em todo momento.

Aos meus orientadores, José Roberto Boisson e Gláucio Lima Siqueira, sempre dispostos a sanar dúvidas e dar recomendações oportunas.

À minha família e amigos que sempre estiveram atentos aos meus progressos.

Ao CNPq, à FAPERJ e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Resumo

Sierra Franco, Cesar Augusto; Lima Siqueira, Gláucio (Orientador); Boisson de Marca, José Roberto (Co-orientador). **Técnicas de aprendizagem para gerência de recursos em redes móveis heterogêneas e auto-organizáveis.** Rio de Janeiro, 2018. 134p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Os sistemas de comunicações móveis atuais vêm enfrentando novos desafios, marcados pelo aumento do uso de novos dispositivos e pela mudança nos padrões de consumo de banda causada pelas aplicações emergentes. É por isso que a indústria de comunicações e a comunidade acadêmica vêm trabalhando tanto nas dificuldades apresentadas nas redes móveis atuais quanto nos desafios técnicos para o desenvolvimento dos esperados sistemas de quinta geração (5G). O grande aumento dos elementos da rede de acesso rádio e a implementação de cenários heterogêneos (macro e pico eNBs, Relay Nodes, etc.) são duas das principais abordagens utilizadas para melhorar a capacidade da rede. No entanto, esse acréscimo de elementos ou, densificação, traz consigo um aumento nos custos e na complexidade nas tarefas de operação e gerenciamento do sistema, já que os novos elementos de rede precisam ser adaptados, configurados e gerenciados continuamente para garantir e aumentar a eficiência da rede, melhorando a qualidade nos serviços oferecidos aos usuários. Este trabalho de pesquisa propõe a inclusão de mecanismos cognitivos, incluindo técnicas de adaptação, nas arquiteturas das redes de acesso móvel. O trabalho propõe igualmente novos mecanismos de auto-organização (*Self Organizing Networks*, SON) para o balanceamento de carga, empregando modelos dinâmicos capazes de tomar decisões inteligentes e aprender a partir de experiências para atingir os objetivos de desempenho desejados.

Palavras-chave

Redes móveis auto-organizáveis; redes heterogêneas; redes cognitivas; balanceamento de carga.

Abstract

Sierra Franco, Cesar Augusto; Lima Siqueira, Gláucio (Advisor); Boisson de Marca, José Roberto (Co-Advisor). **Machine learning techniques for resource management in mobile self-organizing networks**. Rio de Janeiro, 2018. 134p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Today's mobile communications systems are facing new challenges, triggered by the increased use of new devices and the growth of bandwidth hungry applications. This is why over the last years, the telecommunication industry and academic communities have been focused on research and development of technologies for the upcoming 5th generation mobile systems (5G). Among the potential candidates, network densification has attracted growing attention as a key mechanism to fulfill the objective proposed in 5G, by increasing the number of radio-base stations (on the coverage area) and introducing an additional layer of low-power access nodes (e.g., Femto, picocells, relay nodes). However, this approach has also posed new challenges in network configuration, management, and optimization tasks to ensure and increase the mobile network efficiency. This research proposes the inclusion of cognitive mechanisms and adaptive techniques in the architectures of mobile radio access networks. This work also proposes new self-organizing (SON) functions for load balancing, enhanced with capabilities of learning from previous experiences to achieve future desired performance goals.

Keywords

Self-organizing networks; heterogeneous networks; cognitive networks; load balance.

Sumário

1	Introdução	17
1.1.	Contexto e motivação	17
1.2.	Objetivos	20
1.2.1.	Objetivos específicos	20
1.3.	Trabalhos relacionados	21
1.3.1.	Arquiteturas cognitivas em redes de comunicações	21
1.3.2.	Balanceamento de carga cognitivo em redes móveis	23
1.4.	Contribuições	25
1.5.	Organização da tese	27
2	Conceitos em redes auto-organizáveis	28
2.1.	Auto-organização nas redes de comunicações	28
2.2.	Escopo e funcionalidades SON nas redes móveis celulares	30
2.2.1.	Auto-configuração	31
2.2.2.	Auto-otimização	32
2.2.3.	Auto-recuperação	32
3	Técnicas de inteligência computacional e aprendizagem de máquina	33
3.1.	Aprendizagem supervisionada por regressão linear multivariada	34
3.2.	Aprendizagem por reforço	36
3.2.1.	Exploração vs. aproveitamento	37
3.2.2.	Processos de decisão de Markov: conceitos gerais	37
3.2.3.	Determinando as políticas ótimas a partir dos valores de $T(s, a, s')$ e $R(s, a, s')$	38
3.2.4.	Métodos de aprendizado	40
3.2.5.	Algoritmo Q-Learning (livre de modelo)	41
4	Arquitetura cognitiva e cooperativa para redes móveis auto-organizáveis	43
4.1.	Descrição geral	43
4.2.	Princípios de desenvolvimento	45

4.2.1. Soluções distribuídas	45
4.2.2. Modularidade	46
4.2.3. Self-Awareness	46
4.2.4. Capacidades de aprendizagem	46
4.2.5. Desenvolvimentos multicamada	47
4.2.6. Capacidades de compartilhamento de informação	47
4.3. Arquitetura funcional	47
4.3.1. Camada de percepção	49
4.3.2. Camada de decisão/aprendizagem	49
4.3.3. Camada de operação	50
4.4. Integração na rede de acesso rádio móvel.	51
4.5. Modelagem e simulação	52
5 Técnicas cognitivas de balanceamento de carga em redes móveis heterogêneas.	56
5.1. Balanceamento de carga em redes móveis heterogêneas: seleção de célula e expansão virtual de cobertura	56
5.2. Caso de estudo e cenário de simulação	57
5.3. Técnica 1: balanceamento de carga cognitivo através de uma abordagem de aprendizagem supervisionado	59
5.3.1. Fase de aprendizagem	60
5.3.1.1. Obtenção dos dados de treinamento	60
5.3.1.2. Extração das características/atributos	61
5.3.1.3. Aplicação dos algoritmos de aprendizagem (extração do modelo)	61
5.3.1.4. Avaliação	62
5.3.2. Seleção do CRE- <i>offset</i>	63
5.3.3. Implementação na arquitetura LTE	63
5.3.4. Avaliação de desempenho	65
5.3.4.1. Modelo extraído	65
5.3.4.2. Perda de pacotes na camada PDCP	67
5.3.4.3. Utilização de recursos na PeNB	69
5.4. Técnica 2: balanceamento de carga cognitivo através de uma abordagem por aprendizagem por reforço	69

5.4.1. Algoritmo Fuzzy Q-Learning	70
5.4.2. Definição de estados, ações e recompensas	73
5.4.2.1. Estados	73
5.4.2.2. Ações/consequentes.	74
5.4.2.3. Função de recompensa	75
5.4.3. Implementação na arquitetura LTE	75
5.4.4. Simulação do processo de aprendizagem	76
5.4.5. Avaliação de desempenho	77
5.4.5.1. Resultados da aprendizagem	78
5.4.5.2. Avaliação das métricas de QoS	79
5.5. Observações finais	82
6 Técnica cognitiva de balanceamento de carga em redes móveis heterogêneas multi-serviço	84
6.1. Expansão virtual de cobertura multi-serviço	86
6.2. Modelos de estimação da qualidade da experiência	87
6.2.1. Modelo de avaliação de QoE para serviços de vídeo	88
6.2.2. Modelo de avaliação de QoE para serviços de voz	89
6.2.3. Modelo de avaliação de QoE para serviços de transferência de arquivos	90
6.3. Mecanismo de aprendizagem autônomo para o balanceamento de carga em redes móveis multi-serviço	90
6.3.1. Formulação do problema de aprendizagem	91
6.3.2. Definição de variáveis de estado e análise dimensional	92
6.3.2.1. Elementos percebidos pela MeNB	92
6.3.2.2. Elementos percebidos pela PeNB	93
6.3.2.3. Análise dimensional	94
6.3.3. Solução de aprendizagem por reforço com estrutura hierárquica	94
6.3.4. Modelagem e processo de aprendizado para cada subtarefa	97
6.3.4.1. Espaço de sub-estados	98
6.3.4.2. Ações	99
6.3.4.3. Funções de recompensa	99
6.3.5. Implementação do mecanismo Fuzzy Q-Learning hierárquico	101
6.3.5.1. Definição dos FIS para cada subtarefa	101

6.3.5.2. Algoritmo FQL multi-serviço	104
6.4. Implementação na arquitetura LTE	108
6.5. Descrição do cenário simulação e a etapa de aprendizagem	110
6.5.1. Descrição do cenário de simulação	110
6.5.2. Modelagem e simulação dos tipos de serviços trafegados na célula	112
6.5.3. Simulação do processo de aprendizagem	113
6.6. Avaliação de desempenho	114
6.6.1. Evolução temporal durante a fase da aprendizagem	114
6.6.2. Avaliação de desempenho na etapa de otimização	116
7 Conclusões e trabalhos futuros	123
7.1. Conclusões	123
7.2. Trabalhos futuros	125
8 Referências bibliográficas	126

Lista de Figuras

Figura 1. Metodologia para o desenvolvimento de funcionalidades SON em sistemas móveis 2G/3G/4G [3]	19
Figura 2. Ciclo cognitivo	19
Figura 3. Operação manual de rede (fonte: [34])	28
Figura 4. Taxonomia para SON [2]	31
Figura 5. Processo de aprendizado de máquina supervisionado	35
Figura 6. Agente com aprendizagem por reforço	37
Figura 7. Processo de Markov	38
Figura 8. Processo de Markov (MDP) visto como uma árvore de busca	38
Figura 9. Fluxo de informação em um dispositivo de acesso com capacidades cognitivas e de auto-organização	44
Figura 10. Fontes de informação para serem utilizadas pela técnica de aprendizagem	45
Figura 11. Arquitetura funcional do agente cognitivo	48
Figura 12. Estrutura interna de um módulo de auto-organização cognitivo	50
Figura 13. Formação de grupos cognitivos	50
Figura 14. Interface multicamada	51
Figura 15. Visão geral do modulo LTE-EPC implementado no simulador NS-3: a) entidades modeladas, b) pilha de protocolos no plano de dados do núcleo de rede e a interface aérea [69]	53
Figura 16. Arquitetura do software desenvolvido para a modelagem e simulação da arquitetura CCSON	54
Figura 17. Iteração entre as ferramentas Matlab e NS-3	55
Figura 18. Técnica " <i>cell range expansion</i> "	57
Figura 19. Cenário de estudo e simulação	58
Figura 20. Diagrama de blocos do processo de aprendizado e seleção de CREO	59
Figura 21. Agente cooperativo associado a MeNB	64
Figura 22. Agente cognitivo associado a PeNB	64
Figura 23. Modelo extraído: gráfico em 3D	66

Figura 24. Modelo extraído: gráfico de contorno	67
Figura 25. Perda de pacotes na camada PDCP média Vs número de UEs ativos na célula	68
Figura 26. FDA empírica para a perda de pacotes média na camada PDCP (40 UEs)	68
Figura 27. Utilização de blocos de recursos na PeNB	69
Figura 28. Pico eNB como agente de aprendizagem	70
Figura 29. Funções de pertinência para a variável de estado referente ao número de usuários na MeNB/PeNB.	74
Figura 30. Funções de pertinência para a variável de estado que representa a qualidade dos UEs na borda da pico célula	74
Figura 31. Funções de pertinência para a variável de saída <i>CREO</i>	75
Figura 32. Agente cooperativo associado a MeNB	76
Figura 33. Agente cognitivo associado a PeNB	76
Figura 34. Relação aprendida entre as variáveis <i>NMUE</i> , <i>SINR5</i> e <i>CREO</i> que produzem o melhor desempenho na célula	78
Figura 35. Relação aprendida entre as variáveis NPUE, <i>SINR5</i> e <i>CREO</i> que produzem o melhor desempenho na célula	79
Figura 36. Perda de pacotes média na camada PDCP vs. Número de UEs ativos na célula	79
Figura 37. FDA empírica para a perda de pacotes na camada PDCP (40 UEs)	80
Figura 38. Vazão média por UE na camada PDCP vs. Número de UEs ativos na célula	80
Figura 39. FDA empírica para a vazão média por UE na camada PDCP (40 UEs)	81
Figura 40. Expansão virtual de cobertura multi-serviço	92
Figura 41. Estrutura hierárquica aplicada à técnica de expansão virtual de cobertura multi-serviço	95
Figura 42. Sequência temporal de subtarefas desencadeadas pela macro-ação: Modificar CRE-OFFSET	97
Figura 43. Definição dos elementos de cada sub-MDP	100
Figura 44. Funções de pertinência e conjuntos <i>Fuzzy</i> para a variável: número de comunicações (<i>radio access bearers</i>) ativas com	

tráfego pertencente à classe de serviços de a) vídeo, b) voz e c) dados	102
Figura 45. Funções de pertinência e conjuntos <i>Fuzzy</i> para a variável: percentagem de utilização de blocos de recursos na MeNB pela classe de serviços <i>k</i>	103
Figura 46. Funções de pertinência e conjuntos <i>Fuzzy</i> para a variável: percentagem de utilização de blocos de recursos na PeNB	103
Figura 47. Funções de pertinência e conjuntos <i>Fuzzy</i> para a variável: qualidade do estado do canal dos usuários da borda da pico-célula com tráfego pertencente a classe de serviços <i>k</i>	103
Figura 48. Funções de exploração	107
Figura 49. Agente cooperativo associado a MeNB	109
Figura 50. Agente cognitivo associado a PeNB.	109
Figura 51. Cenário de simulação	111
Figura 52. Descrição do processo em cada episódio de treinamento durante a fase de aprendizagem	114
Figura 53. Evolução temporal dos valores de <i>CRE-offsets</i> de vídeo, voz e dados.	115
Figura 54. Evolução temporal da QoE média por classe de serviço como consequência na mudança dos <i>CRE-offsets</i> (vídeo, voz e dados)	116
Figura 55. Vazão total da célula vs. número de UEs ativos	118
Figura 56. QoE média na célula por usuário Vs. Número de UEs ativos na célula	119
Figura 57. QoE média por tipo de serviço para o cenário com 45 UEs ativos na célula	119
Figura 58. FDA empírica para a QoE média por usuário com serviço de vídeo (cenário com 45 UEs)	120
Figura 59. FDA empírica para a QoE média por usuário com serviço de voz (cenário com 45 UEs)	121
Figura 60. FDA empírica para a QoE média por usuário com serviço de dados (cenário com 45 UEs)	121

Lista de Tabelas

Tabela 1. Parâmetros de simulação para o E-UTRAN (Fonte: [72])	58
Tabela 2. Resumo das técnicas cognitivas desenvolvidas para o balanceamento de carga em redes móveis heterogêneas	82
Tabela 3. Glossário das principais variáveis utilizadas neste capítulo	85
Tabela 4. Escala de qualidade MOS	87
Tabela 5. Coeficientes da função MOS_video para as categorias de vídeo: movimentos suaves, moderados e rápidos	88
Tabela 6. Resumo dos aspectos principais da técnica de balanceamento cognitivo multi-serviço	110
Tabela 7. Função de probabilidade para a distribuição de tráfego nos cenários de treinamento	111
Tabela 8. Parâmetros de simulação do E-UTRAN [72]	112
Tabela 9. Condições de tráfego utilizadas nos cenários de teste	117

Abreviaturas

3GPP	<i>3rd Generation Partnership Project</i>
CCSON	<i>cognitive and cooperative for self-organizing mobile networks</i>
CQI	<i>Channel quality information</i>
CRE	<i>Cell Range Expansion</i>
CREO	<i>Cell Range Expansion Offset</i>
eNBs	<i>Evolved Node B</i>
EO	<i>Elemento de operação</i>
EP	<i>Elemento de percepção</i>
FIS	<i>Fuzzy Inference System</i>
FQL	<i>Fuzzy Q-Learning</i>
FR	<i>Frame Rate</i>
KPI	<i>Key performance indicator</i>
LTE	<i>Long Term Evolution</i>
MDP	<i>Markov Decision Process</i>
MeNB	<i>Macro eNB</i>
MOS	<i>Mean opinion score</i>
NP	<i>Network parameter</i>
NRAB	<i>Number of radio access bearers</i>
NS-3	<i>Network Simulator 3</i>
PDCCP	<i>Packet Data Convergence Protocol</i>
PeNB	<i>Pico eNB</i>
PER	<i>packet error rate</i>
QoE	<i>quality of experience</i>
RL	<i>Reinforcement Learning</i>
RSRP	<i>Reference Signal Received Power</i>
SBR	<i>sender bit rate</i>
SL	<i>Supervised learning</i>
SOM	<i>Self-organizing module</i>
SON	<i>Self Organizing Networks</i>
SSE	<i>sum square error</i>
SSR	<i>sum of squares of the regression</i>

1 Introdução

Este capítulo, dividido em cinco seções, apresenta os conceitos que foram essenciais para o desenvolvimento desta tese. Inicialmente, descreve-se o contexto do estudo e os aspectos que motivaram a pesquisa. A seguir, apresentam-se os objetivos geral e específicos pretendidos. Posteriormente, apresenta-se uma breve descrição dos trabalhos relacionados com esta tese e a contribuições desses artigos. Finalmente, é descrito como o texto da tese está organizado.

1.1. Contexto e motivação

A evolução das redes móveis tem sido caracterizada pela constante introdução de novas tecnologias, dispositivos e arquiteturas de rede, com o objetivo de acompanhar o aumento no número de usuários e as mudanças nos padrões de consumo de banda devido às aplicações emergentes. Uma das principais abordagens utilizadas para melhorar a capacidade da rede é o aumento ou densificação do número de elementos na rede de acesso via rádio e a implementação de cenários heterogêneos (macro/pico-eNB, *Relay Nodes*, etc.) [1]. No entanto, este incremento de dispositivos de acesso rádio trouxe consigo o aumento nos custos e na complexidade nas tarefas de operação/gerenciamento do sistema.

Para enfrentar estes desafios, a aliança para redes móveis de próxima geração (*Next Generation Mobile Networks Alliance, NGMN*), introduziu o conceito de redes auto-organizáveis (*Self Organizing Networks, SON*) como uma das principais abordagens para otimizar o desempenho da rede enquanto se procura reduzir os seus custos operacionais. A auto-organização (*Self Organization, SO*) é um conceito inspirado na natureza, onde sistemas biológicos apresentam um comportamento organizado, observando, aprendendo e se adaptando ao meio a fim de atingir objetivos desejados. Esse conceito de SO tem sido estudado e aplicado em diversas áreas, entre elas: a biologia, as ciências da

computação e a cibernética. Na área das redes de comunicações, SO tem sido estudada principalmente em redes de sensores e redes ad-hoc. Já na área dos sistemas móveis celulares, o *3rd Generation Partnership Project* (3GPP) tem identificado a necessidade de introduzir as ditas funcionalidades para os seus padrões da tecnologia *Long Term Evolution* (*LTE* e *LTE-Advanced*), procurando fornecer à rede inteligência e adaptabilidade para automatizar funcionalidades relacionadas com a configuração, otimização e recuperação de falhas.

Nos últimos anos, o uso de SON nas redes de acesso via rádio tem atraído o interesse da indústria de comunicações e da comunidade acadêmica, levando à realização de vários estudos em temas relacionados com a otimização de cobertura, capacidade, balanceamento de carga e mitigação de interferências [2]. Destaca-se que a abordagem predominante nesses estudos é o desenvolvimento dos mecanismos adaptativos e autônomos usando técnicas de otimização e soluções analíticas. No entanto, apesar desses avanços, diversos trabalhos apontam ainda para carências das funções SON para enfrentar os desafios apontados pelos próximos sistemas de quinta geração (5G) [3].

Pontualmente, neste trabalho estamos interessados nos casos em que as infraestruturas de acesso rádio mudem por causa da introdução não planejada de elementos de acesso, tais como *small-cells*, *relay nodes* ou outro tipo de estações radio-base de baixa potência. Este tipo de cenários heterogêneos se caracteriza por apresentar padrões de interferência imprevisíveis, causados pelo posicionamento aleatório dos elementos de acesso, apresentando dinâmicas complexas que são difíceis de modelar analiticamente [4].

Atualmente, as funcionalidades de auto-organização sustentam seu desenvolvimento em metodologias que pressupõem um conhecimento parcial ou total do problema, a partir de informações coletadas em registros de reclamações de clientes, coleta de dados ou relatórios provenientes do centro de operação/manutenção da rede (Figura 1) [3]. Com isso, a implementação das funções SON podem não acompanhar eficientemente as mudanças na rede.

Nesse sentido, vários estudos [2][3][5] sugerem que as funcionalidades de auto-organização podem ser melhoradas a partir da inclusão de capacidades de aprendizagem (abordagens cognitivas), possibilitando um incremento no nível de autonomia nos mecanismos de auto-organização. Dessa forma, as redes auto-organizáveis não serão vistas apenas como redes adaptativas (sistemas que podem

mudar seus parâmetros de operação em resposta as mudanças no ambiente) ou autônomas (redes adaptativas sem intervenção humana ou externa), mas também como sistemas cognitivos com a possibilidade de aprender e melhorar seu desempenho com base na experiência adquirida no passado [2].

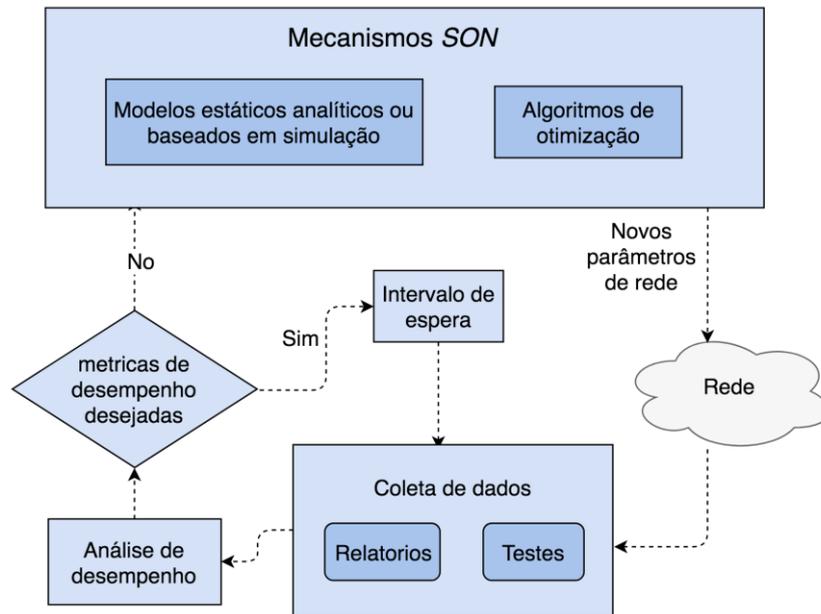


Figura 1. Metodologia para o desenvolvimento de funcionalidades SON em sistemas móveis 2G/3G/4G [3]

Essas abordagens seguem geralmente o conceito do ciclo cognitivo (Figura 2) proposto por Mitola [6], com o objetivo de integrar capacidades de percepção, adaptação e aprendizagem em sistemas de comunicações, permitindo uma contínua adaptação e otimização dos parâmetros da rede conforme às alterações do ambiente.

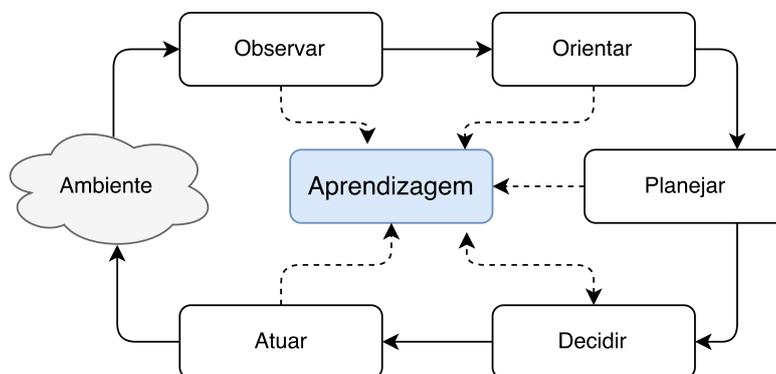


Figura 2. Ciclo cognitivo

Nos últimos anos, vários estudos [3][7][8] têm usado técnicas de inteligência computacional e aprendizado de máquina, buscando incrementar a autonomia das funcionalidades SON. Embora os resultados sejam promissores, este ainda é um campo em desenvolvimento, sem um consenso ou padronização definido sobre como deve ser abordada a integração de capacidades de aprendizagem nas funcionalidades SON.

Nessa perspectiva, a motivação desta tese é aprofundar no estudo de arquiteturas e processos cognitivos (centrados na aprendizagem) em redes de acesso móvel, visando contribuir com o desenvolvimento de novas técnicas de auto-organização que melhorem o desempenho do sistema através da experiência.

1.2. Objetivos

O objetivo geral desta tese é desenvolver mecanismos cognitivos/cooperativos de auto-organização em redes móveis heterogêneas, e aplicá-los num problema específico de balanceamento de carga em cenários multi-serviço. Com isso, pretende-se mostrar como esses mecanismos podem ser incluídos nas arquiteturas de acesso rádio, propiciando a integração de diversas técnicas de inteligência computacional no contexto das redes móveis celulares.

1.2.1. Objetivos específicos

- Identificar os princípios fundamentais a serem considerados no desenvolvimento de mecanismos cognitivos.
- Desenvolver uma arquitetura para integrar mecanismos cognitivos que apoiem as funções de auto-organização nas redes de acesso móvel, tornando-a reutilizável e expansível para outras soluções de auto-organização.
- Desenvolver técnicas de auto-organização com diferentes abordagens no seu processo de aprendizado (supervisionado e aprendizagem por reforço), tendo como foco o problema de balanceamento de carga entre macro e pico células.

- Desenvolver uma técnica de balanceamento de carga auto-organizável, sustentada em processos de aprendizagem para redes móveis multi-serviço (vídeo, voz e transferência de arquivos).

1.3. Trabalhos relacionados

Na literatura têm sido apresentadas diversas propostas cognitivas em redes sem fio, variando seu foco de pesquisa desde a definição conceitual nas arquiteturas funcionais, até implementações para solucionar problemas específicos de rede. Apresentam-se a seguir os estudos considerados mais relevantes com relação aos aspectos tratados nesta tese, tanto estudos sobre arquiteturas para incluir técnicas de aprendizagem bem como estudos sobre balanceamento de carga utilizando abordagens cognitivas.

1.3.1. Arquiteturas cognitivas em redes de comunicações

Os primeiros conceitos na área de sistemas cognitivos de comunicações surgiram do trabalho sobre rádio cognitivo de Mitola e Maguire [6], sendo posteriormente estendidos para a resto da rede por Thomas, DaSilva e MacKenzie no seu trabalho intitulado Redes Cognitivas [9]. Nesse trabalho, Thomas e colaboradores tiveram como objetivo integrar as capacidades de percepção, adaptação e aprendizagem para atingir objetivos de desempenho fim-a-fim. Para isso, identificaram a necessidade de implementar funcionalidades de *software* que permitam obter informações (“*sensing*”) e modificar parâmetros na pilha de protocolos nos elementos da rede, conforme os resultados dos mecanismos e processos cognitivos. Os autores implementaram a abordagem desenvolvida num caso de estudo do ajuste de potências de transmissão em redes sem-fio. Eles mostraram que a solução baseada em processos cognitivos proporcionou melhores soluções quando comparada com uma técnica tradicional baseada em heurísticas.

De forma geral, as arquiteturas e modelos conceituais para a implementação de sistemas cognitivos, seguem os princípios da computação autônoma propostos pela IBM e as abordagens de gerenciamento autônomo [10]. Desta

forma, os sistemas gerenciados são associados com componentes de *software* que integram mecanismos de sensoriamento, raciocínio e atuação.

O modelo proposto por Thomas e colaboradores [9] se compõe por três camadas horizontais. A camada superior é responsável pela interpretação dos requisitos dos usuários em termos de objetivos entendíveis pelos mecanismos cognitivos. A camada intermedia implementa os algoritmos de decisão, enquanto que a última camada é a interface de *software* que permite a modificação dos parâmetros de operação dos elementos de rede.

De forma similar, Mähoänen e colaboradores [11] propuseram uma arquitetura para rádio cognitiva com funcionalidades de gerenciamento de recursos (*cognitive resource management*), e otimização de parâmetros em toda a pilha de protocolos rádio de um sistema de comunicações sem-fio. Esta arquitetura conceitual considera o uso de diversas técnicas para a criação dos modelos de decisão, tais como, redes neurais, algoritmos genéticos, redes bayesianas, entre outros. Os autores concluem que ampliando o escopo da pesquisa de rádio cognitivo para a totalidade da pilha de protocolos, possibilita-se a obtenção de melhores resultados quando comparado com técnicas focadas apenas na análise da camada física e nos problemas de espectro de frequências.

Um dos principais desafios no desenvolvimento de arquiteturas cognitivas consiste em codificar e armazenar a informação que representa o “conhecimento” adquirido pela rede [12]. Nesse sentido, alguns trabalhos têm focado no uso de formalismos e técnicas específicas para tal fim. Por exemplo, em [13], Baldo e colaboradores sugeriram o uso da lógica *Fuzzy* para representar a informação multicamada e para implementar os sistemas de inferência responsáveis pela otimização dos recursos rádio, independentemente das aplicações e/ou tecnologias de rádio empregadas. De forma similar, Al-Saadi e colaboradores [14][15] propuseram uma arquitetura baseada em abordagens semânticas tanto para o armazenamento da informação quanto para os mecanismos de inferência. Especificamente, a arquitetura proposta nesse trabalho utiliza ontologias e sistemas de inferência *Fuzzy* para criar um raciocínio baseado em regras e controlar diversos parâmetros de rede de varias tecnologias sem-fio simultaneamente.

Zorzi e colaboradores descrevem uma arquitetura para redes cognitivas que denominaram COBANETS (*COgnition-BAsed NETworkS*) [16][17]. Os autores

propuseram o uso de técnicas de redes neurais profundas não supervisionadas como um dos principais elementos para o desenvolvimento de novos sistemas de comunicações cognitivos.

Do nosso ponto de vista, a ideia de definir um tipo de técnica de inteligência computacional para representar/armazenar o “conhecimento” da rede pode ser efetiva na solução de um problema específico. Porém, torna o sistema dependente do tipo de mecanismo utilizado. Nesse sentido, consideramos que é importante desenvolver arquiteturas que permitam a coexistência de múltiplas abordagens de aprendizagem de máquina e técnicas de otimização.

Do ponto de vista dos processos de aprendizagem, estas características têm sido abordadas a partir do uso de técnicas de inteligência computacional, mostrando sua efetividade para incrementar a autonomia das funcionalidades SON [3][7][18]. Em um trabalho recente, Klaine e colaboradores [8] apresentaram uma revisão da literatura sobre a aplicação de algoritmos de aprendizagem de máquina em redes móveis celulares, classificando os trabalhos conforme o tipo de algoritmo de aprendizagem utilizado: supervisionado, não supervisionado, aprendizagem por reforço, redução de dimensionalidade e transferência de aprendizado. Nesse trabalho, os autores destacam os benefícios do uso do paradigma baseado em técnicas de aprendizagem, fornecendo orientações para futuras pesquisas.

1.3.2. Balanceamento de carga cognitivo em redes móveis

Na literatura têm sido apresentadas diversas contribuições em relação ao balanceamento de carga em redes móveis [19], sendo este tipo de problema usualmente modelados como problemas de controle de acesso, controle de processos de *handover*, e/ou ajuste de potências de transmissão das estações rádio base. Estes trabalhos têm sido abordados com uma visão analítica a partir de modelos fundamentados em variações da fórmula de capacidade de Shannon para determinar a capacidade do sistema [20], porém, ignorando alguns aspectos importantes que afetam o desempenho de um sistema móvel, tais como o agendamento de pacotes na camada MAC, e outros relacionados com as camadas superiores [4].

Nesse sentido, as abordagens baseadas em aprendizagem de máquina procuram construir elementos de apoio à decisão a partir de observações e experiências do sistema que permitam posteriormente realizar previsões e ajustes para otimizar o desempenho da rede.

A partir do uso de técnicas de aprendizagem, diversos trabalhos têm focado no controle dos parâmetros relacionados com *handover* para realizar um balanceamento de carga adaptativo [21]–[23]. Entre esses, Atayero e colaboradores apresentaram um modelo *Neuro-Fuzzy* para ajustar o parâmetro de histerese utilizado durante o procedimento de *handover* para forçar aos usuários na borda da célula selecionar estações base vizinhas com uma menor carga de tráfego [22]. Nesse trabalho, o principal indicador de desempenho utilizado para realizar os ajustes é o índice de distribuição de carga (*Load Distribution Index*, LDI), que mede o grau de balanceamento de carga na totalidade da rede. O problema ao expandir esta solução para um cenário heterogêneo decorre das diferentes potências de transmissão e áreas de cobertura entre as macro e pico células. Ao tentar forçar uma distribuição de carga igualitária entre os dois tipos de células (sem ter em conta outros indicadores de qualidade), resulta em um incremento do número de *handovers* prematuros desde as macro-células para as bordas das pico-células que ainda apresentam altas interferências.

Em outro estudo, Kudo e Ohtsuki [24] apresentaram uma técnica de balanceamento de carga entre macro e pico-células usando uma técnica de expansão virtual de cobertura (*Cell Range Expansion*, CRE) para as pico células. Nesse trabalho, os autores utilizaram uma abordagem baseada em aprendizagem por reforço para que os terminais de usuário aprendam, por tentativa-erro, o *offset* utilizado pela técnica CRE, minimizando o número de interrupções nas comunicações e melhorando conseqüentemente o desempenho da célula.

De forma similar, Mwanje e Mitschele-Thiel, desenvolveram um algoritmo de aprendizado para ajustar dinamicamente *offsets* individuais (utilizados pelas técnicas de *handover*) entre a célula origem e as células vizinhas [23]. O ajuste do parâmetro foi realizado por passos fixos, aplicando aprendizagem por reforço para determinar o melhor valor de ajuste conforme o estado da célula.

Outras soluções têm implementado funcionalidades de autoaprendizagem para ajustar parâmetros de configuração das antenas nas estações rádio-base [25]–[27]. Dentre elas, Tomforde e colaboradores propõem um algoritmo que permite

as estações rádio-base modificar os ângulos de inclinação das antenas conforme o tráfego da célula [27]. Para isso, a célula congestionada analisa dados históricos (armazenados previamente na execução do algoritmo) para prever quais células vizinhas (e valores de inclinação de antenas) poderiam aliviar a carga de tráfego.

Todas as abordagens anteriores fornecem vantagens no balanceamento de carga em diferentes configurações e topologias da rede de acesso móvel. No entanto, a maioria destes trabalhos têm sido testados apenas para cenários simplificados com distribuições de tráfego constante e/ou com tipos de tráfego homogêneos. No melhor do nosso conhecimento, poucos trabalhos levam em consideração cenários multi-serviço quando utilizando abordagens de aprendizado de máquina.

1.4. Contribuições

Considerando o mencionado acima, este trabalho de pesquisa propõe três contribuições principais:

1. O desenvolvimento de uma arquitetura cognitiva e cooperativa que propicia o uso de técnicas de aprendizagem para o gerenciamento de recursos em redes móveis heterogêneas e auto-organizáveis.

Ao nível das funcionalidades de auto-organização, esta arquitetura especifica os componentes, estruturas e relacionamentos necessários para integrar mecanismos cognitivos/cooperativos nas arquiteturas móveis. O principal objetivo é prover os elementos da rede de acesso com capacidades de autoaprendizagem, de tal forma que eles consigam criar/adaptar dinamicamente modelos ou políticas para sustentar as funcionalidades *SON*. Esta arquitetura foi projetada para operar de forma distribuída por meio de agentes associados aos elementos de acesso rádio. Desta forma, favorecemos a criação de grupos cognitivos para contribuir na otimização na rede de acesso. Esta abordagem também teve como objetivo evitar a dependência em um tipo de formalismo ou técnica específica. Para isso, os detalhes das funcionalidades *SON*-cognitivas foram encapsulados em módulos independentes próprios dos

agentes associados aos elementos de acesso, possibilitando a coexistência de múltiplas técnicas de aprendizagem de máquina.

2. O desenvolvimento de técnicas de balanceamento de carga cognitivas em redes móveis auto-organizáveis.

Neste trabalho, foi abordado o problema do balanceamento de carga em redes móveis entre macro e pico-células [28]. As técnicas aqui desenvolvidas, integram características de aprendizagem que permitem ao sistema criar modelos/políticas automaticamente para ajustar seus parâmetros de operação conforme o estado da célula. Estas técnicas são fundamentadas na otimização do desempenho em termos de qualidade de serviço (Capítulo 5) e sua possível aplicação em cenários multi-serviço (Capítulo 6), levando em consideração métricas de desempenho centradas na qualidade percebida pelos usuários (QoE).

3. O desenvolvimento de um *software* para a modelagem, simulação e validação de funcionalidades de auto-organização sustentadas em técnicas de aprendizagem de máquina.

O *software* desenvolvido neste trabalho teve como principal característica a separação funcional dos processos de simulação de rede e a implementação dos mecanismos SON-cognitivos. Especificamente, os cenários da rede de acesso móvel (tecnologia *LTE*) foram executados no simulador NS-3 [29] [30], enquanto que a estrutura funcional e os processos matemáticos relacionados com os agentes cognitivos foram implementados na ferramenta Matlab. O *software* criado facilitou o processo de desenvolvimento e validação dos algoritmos de aprendizagem relacionados às funcionalidades de auto-organização.

Publicações. As contribuições mencionadas levaram à publicação dos seguintes artigos:

- “*A Cognitive and Cooperative SON Framework for 5G Mobile Radio Access Networks*”. Na conferência: *IEEE Globecom*, dezembro de 2016 [31].

- “Balanceamento de carga cognitivo em redes móveis heterogêneas e auto-organizáveis”, no XXXIV simpósio brasileiro de telecomunicações e processamento de sinais, SBrT’2016 [32].
- “*Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach*”. Na *7th IEEE Latin-American Conference on Communications (LATINCOM)*, 2015 [33].

1.5. Organização da tese

Esta tese está organizada como segue. No Capítulo 2 são apresentados os conceitos associados com a auto-organização e sua introdução nas redes de comunicações (*Self Organizing Networks*), focando a atenção no escopo e nas funcionalidades SON recomendadas para as redes móveis celulares. No Capítulo 3 são descritas as técnicas de inteligência computacional e aprendizado automático utilizadas de forma exploratória para apoiar os processos cognitivos dentro dos mecanismos de auto-organização a serem desenvolvidos. No Capítulo 4 é detalhado o processo de desenvolvimento da arquitetura projetada para a integração de processos de aprendizagem automático visando apoiar as funções de auto-organização nas redes de acesso móvel. Já no Capítulo 5, é apresentado o desenvolvimento de técnicas de balanceamento de carga cognitivo através de duas abordagens: supervisionada e aprendizagem por reforço para cenários com tráfego homogêneo (um único tipo de serviço). No Capítulo 6, detalha-se a implementação da técnica de balanceamento de carga para cenários multi-serviço (vídeo, voz e transferência de arquivos) e os respectivos resultados obtidos. Esta implementação leva em consideração a influência de diferentes classes de serviço sobre o desempenho da rede e a qualidade de experiência (QoE - *quality of experience*) percebida pelos usuários. Finalmente, no Capítulo 7 apresentam-se as conclusões e proposta de trabalhos futuros.

2 Conceitos em redes auto-organizáveis

2.1. Auto-organização nas redes de comunicações

A constante evolução e crescimento das atuais redes de comunicações móveis vêm tornando as atividades associadas a seu gerenciamento em tarefas complexas, desafiadoras e também com uma operação mais dispendiosa. Isso se deve principalmente ao grande e crescente número de elementos de rede instalados e à interdependência existente entre seus parâmetros de operação. Usualmente, estas tarefas (p.ex., planejamento, configuração, otimização) são executadas de forma centralizada e sob a supervisão de operadores humanos (Figura 3). Isso exige um alto grau de conhecimentos e de especialização, além de estar sujeita a erros e atrasos na atualização dos parâmetros de funcionamento, em comparação com a rapidez nas variações nas condições de operação da rede (topologias, tráfego, etc.). Nesse sentido, as funcionalidades SON podem ser utilizadas para reduzir custos de operação relacionados com o trabalho manual e minimizando os tempos de resposta.

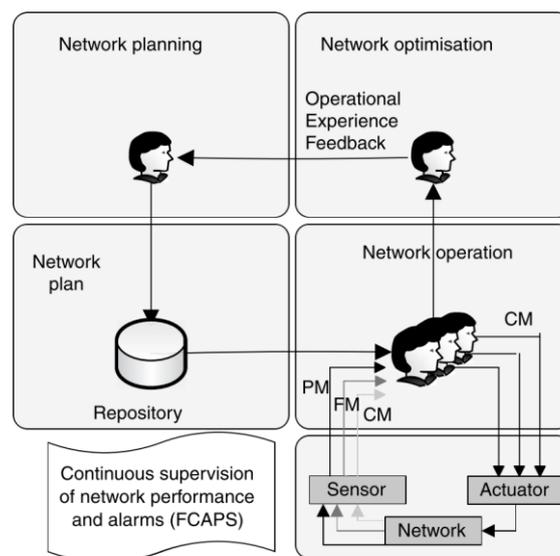


Figura 3. Operação manual de rede (fonte: [34])

Várias iniciativas e conceitos têm sido investigados nesta área, mas todos eles convergem para o surgimento de uma nova geração de equipamentos, de redes e de serviços inteligentes com capacidades e comportamentos auto-adaptativos. Na literatura podem se encontrar diversas terminologias que fazem referência a esta área de pesquisa tais como: comunicações autônomas (*Autonomic Communication, AC*), redes autônomas (*Autonomic Networks, AN*), gerenciamento autonômico de redes (*Autonomic Network Management, ANM*) e redes auto-gerenciáveis (*Self-Managed Networks, SMN*) entre outras. Todos estes termos apresentam diferenças em seus focos de pesquisa, mas todos eles apontam para um objetivo em comum: introduzir capacidades auto-adaptativas nas redes de comunicações e evitar as intervenções ao máximo por parte dos operadores humanos quanto for possível [34].

Por outra parte, a auto-organização (*self organization, SO*) é um conceito inspirado na natureza, onde certos sistemas biológicos apresentam um comportamento organizado, observando, aprendendo e se adaptando ao meio a fim de atingir objetivos desejados. Este conceito tem sido definido e estudado em diversas áreas, incluindo a biologia, ciências da computação e a cibernética. Já na área dos sistemas de comunicações podem ser encontrados vários trabalhos que apresentam definições e conceitos para a adoção de SO.

Spilling e colaboradores [35] introduziram a ideia que SO deveria ser visto como uma série de funcionalidades adaptativas, onde a rede tivesse a capacidade de detectar mudanças no cenário de operação e tomar decisões inteligentes para minimizar os efeitos dessas mudanças. Por outro lado, Yanmaz et al. [36], definiram SO desde a perspectiva das redes cooperativas, onde os nós trabalham conjuntamente em resposta às mudanças no ambiente para atingir objetivos específicos. Neste sentido, e tal como se afirma no trabalho de Prehofer e Bettstetter [37], o comportamento adaptativo resultante do sistema em geral se encontrará baseado nas interações locais entre as entidades que compõem o sistema. Por sua parte, Elmenreich e Meer discutiram em [38] quais seriam as estratégias para lidar com a complexidade associada ao desenvolvimento de sistemas dinâmicos, autônomos e auto-organizáveis. Suas ideias ressaltam o conceito de SON como um conjunto de entidades que obtêm um comportamento global do sistema graças às interações locais entre estas entidades, sem a necessidade de um controle centralizado. Na prática e do ponto de vista das

operadoras, SON é uma camada adicional de inteligência do *software*, que permite automatizar tarefas de gerenciamento, configuração e otimização de forma rápida e com uma mínima intervenção de operadores humanos [39].

Nos conceitos mencionados, destaca-se a recorrência das palavras chave: adaptabilidade e autonomia. No entanto, cabe destacar a relevância de mais uma característica em SO, esta consiste na capacidade de aprendizado das suas entidades, permitindo que estas interajam com o meio e aprendam das experiências passadas.

Nessa ordem de ideias, as redes auto-organizáveis não devem ser vistas apenas como sinônimos das redes adaptativas (sistemas que podem mudar seus parâmetros de operação em resposta as mudanças no ambiente) ou de redes autônomas (redes adaptativas sem intervenção humana ou externa), mas também como sistemas cognitivos com a possibilidade de aprender e melhorar seu desempenho com base em realimentações de ações realizadas previamente [2].

2.2.

Escopo e funcionalidades SON nas redes móveis celulares

Nas redes de comunicações, SO tem sido estudado principalmente em redes de sensores sem fio [40] e redes ad-hoc [41]. No entanto, no contexto das redes móveis celulares, SO é ainda um tema relativamente novo e em desenvolvimento. Em [42], a NGMN (*Next Generation Mobile Networks*) identificou uma série de casos de uso e recomendações para os operadores, com o objetivo de fornecer orientações gerais no desenvolvimento de funcionalidades SON, abrangendo vários aspectos tais como: expansão da cobertura, otimização de capacidade, otimização de QoS e eficiência energética. Posteriormente, o 3GPP iniciou os trabalhos de padronização das funcionalidades de auto-organização e auto-otimização para LTE em suas versões 8 e 9 [43]. Estes padrões procuram fornecer inteligência e automatização aos processos de gerenciamento para automatizar tarefas de configuração. Este esforço tem continuado nas *Releases* mais recentes do LTE, com melhoras adicionais, permitindo o controle e redução de interferências, balanceamento de carga e otimização de cobertura.

Na literatura [2], as técnicas e aplicações SON relevantes às comunicações móveis celulares têm sido categorizados conforme os objetivos e fases no ciclo de vida da rede. Desta forma, são destacadas as seguintes categorias:

autoconfiguração (fase de desenvolvimento), auto-otimização (fase de operação) e autocorreção (fase de manutenção). A Figura 4 ilustra a taxonomia mais comumente utilizada por SON incluindo os principais casos de uso para cada uma delas.

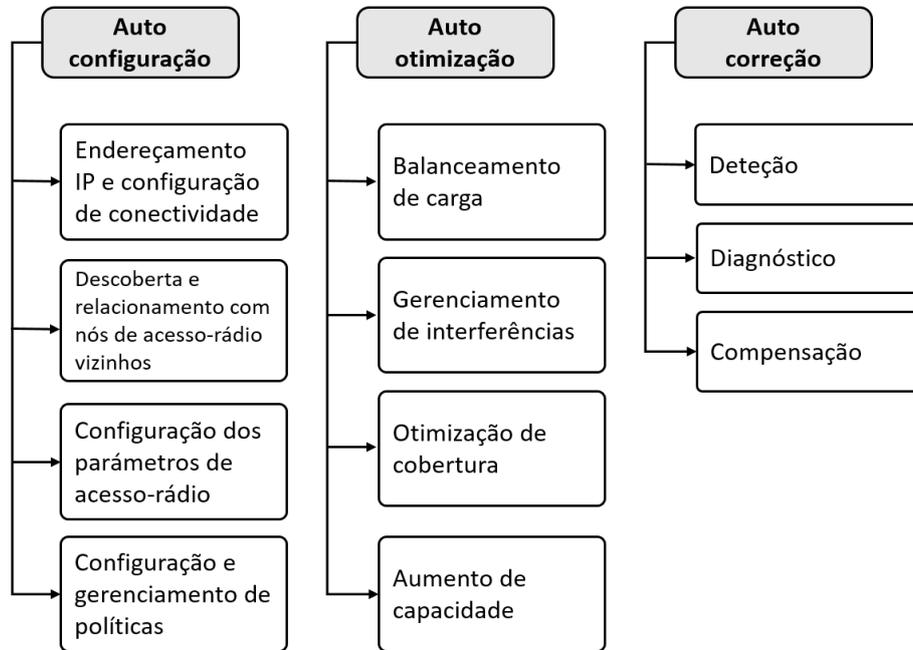


Figura 4. Taxonomia para SON [2]

2.2.1. Auto-configuração

Na rede de acesso sem fio, a implantação de novos elementos de rede (estações base, nós de retransmissão, etc.) ou alterações de *hardware/software* podem causar um esforço considerável de replanejamento e configuração. Portanto, o objetivo das funcionalidades SON de autoconfiguração estão focadas em reduzir a intervenção humana no processo de instalação, fornecendo funcionalidades “*plug and play*” nas estações rádio-base. Entre as principais funcionalidades encontradas nesta categoria, podem se mencionar a autoconfiguração dos endereços IP, a geração automática da lista de estações vizinhas e a configuração automática dos parâmetros do acesso rádio. Na literatura podem ser encontrados diversos artigos e documentos [44] [45] [46] que aprofundam nos conceitos e requisitos para o desenvolvimento das funcionalidades dentro desta categoria.

2.2.2. Auto-otimização

Após a fase de configuração inicial, é preciso otimizar continuamente os parâmetros de funcionamento do sistema para garantir um desempenho eficiente da rede bem como o cumprimento dos objetivos de qualidade de serviço determinados pela operadora. Esta necessidade de otimização automática e contínua nas futuras redes móveis têm sido manifestadas em [43], sendo possível encontrar vários trabalhos nesta área em [47]. Dentre as principais funções SON de auto-otimização podem se destacar o balanceamento de carga, o gerenciamento de interferências e a otimização da capacidade e cobertura.

2.2.3. Auto-recuperação

Existe uma grande variedade de causas que podem provocar o mau funcionamento das redes móveis, causas estas que podem ser nos equipamentos ou ainda desastres naturais. Estas anomalias são gerenciadas de forma manual e em alguns casos operadores humanos precisam se mobilizar até a localização do dispositivo para solucionar o problema. Portanto, o uso de aplicações SON de autocorreção permitiria reduzir o esforço e tempo de manutenção na solução de problemas. A 3GPP identificou em [48] os conceitos, requisitos e principais casos de uso de autocorreção em redes móveis, definindo funcionalidades básicas referentes ao monitoramento, diagnóstico e compensação.

3

Técnicas de inteligência computacional e aprendizagem de máquina

A inteligência computacional é um conjunto de metodologias e abordagens computacionais inspiradas na natureza para lidar com problemas complexos. Isso é feito imitando aspectos do comportamento humano, tais como a aprendizagem, percepção, raciocínio, evolução e adaptação [49], procurando o desenvolvimento de agentes inteligentes (p.e. programas informáticos, sistemas computacionais, etc.) capazes de perceber o ambiente e atuar de maneira autônoma para atingir seus objetivos e metas [50]. Estes agentes são considerados inteligentes, pois o processo de tomada de decisões é caracterizado por uma das características a seguir: 1) os agentes são cientes das consequências de suas ações (processos de inferência) e/ou 2) eles são capazes de aprender a partir das experiências passadas.

A realização de processos de inferência torna um agente capaz de prever os possíveis resultados das suas ações (estados futuros) a partir de um estado inicial, encontrando soluções ou planos “inteligentes” (próximas ações) que o levem a atingir os objetivos desejados. Dentre os principais tipos de problemas trabalhados no campo da inteligência artificial destacam-se: os problemas de busca e planejamento (*Search and planning*) [50], os problemas de obedecer restrições (*constraints satisfaction problems*) [51], e processos de decisão de Markov (*Markov Decision processes*) [52].

Por outro lado, as técnicas de aprendizagem autônomas têm como objetivo compreender os mecanismos computacionais para alcançar um melhor desempenho, possibilitando a construção de modelos a partir de dados de treinamento ou experiências. As técnicas mais comuns nesta área de estudo se concentram na aquisição de conhecimento (criação de modelos) para realizar tarefas de classificação ou regressão. As tarefas de classificação envolvem a atribuição de um novo elemento de entrada a um dentre um conjunto finito de classes, enquanto a regressão prediz um valor conforme os novos elementos ou dados de entrada. Por sua vez, os algoritmos utilizados para realizar estas tarefas

podem ser divididos em algoritmos de aprendizado supervisionados (*supervised learning*) [53] e não supervisionados (*unsupervised learning*). Na primeira classe de algoritmos, encontram-se aqueles que precisam que o conjunto de dados de treinamento utilizados seja associado a uma classe ou rotulado por um valor “correto”. Dentre os principais exemplos de algoritmos nesta categoria podem ser mencionados: *Redes Neurais* [54], *Máquinas de Vetores Suporte (Support Vector Machines)* [55], *Redes Bayesianas* [56], *Regressão logística* e *Regressão linear* [57].

Já nos algoritmos de aprendizado não supervisionado, os dados de treinamento não contêm informações sobre os rótulos associados, portanto, estes algoritmos tentam encontrar padrões e estruturas ocultas nos dados não etiquetados. Dentre alguns dos algoritmos nesta categoria podem se mencionar: *clustering* [57], *density estimation* [58] e *Self Organizing Maps* [59].

A seguir são apresentadas em detalhe as duas técnicas de aprendizagem utilizadas até o momento no presente estudo: aprendizagem supervisionada por regressão linear multivariada e aprendizagem por reforço.

3.1.

Aprendizagem supervisionada por regressão linear multivariada

As técnicas de aprendizado de máquinas empregam um princípio de inferência denominado de indução, no qual é possível obter conclusões genéricas a partir de um conjunto particular de exemplos ou experiências. Especificamente no caso da aprendizagem supervisionada é fornecida uma referência do objetivo a ser alcançado, isto é, um treinamento com o conhecimento do ambiente.

Para isso, os algoritmos supervisionados utilizam um conjunto de dados de treinamento composto com entradas representadas por um vetor \bar{x} (de atributos) e uma saída desejada y . O objetivo dos ditos algoritmos é obter um modelo preditivo, inferindo uma função hipótese $h(\bar{x})$ que retorne uma boa predição de saída \hat{y} [53].

A Figura 5 ilustra o processo seguido pelas abordagens utilizadas na aprendizagem supervisionada, que consiste nos seguintes passos: obtenção dos dados de treinamento, extração das características/atributos, aplicação dos algoritmos de aprendizado para a extração do modelo e avaliação dos resultados.

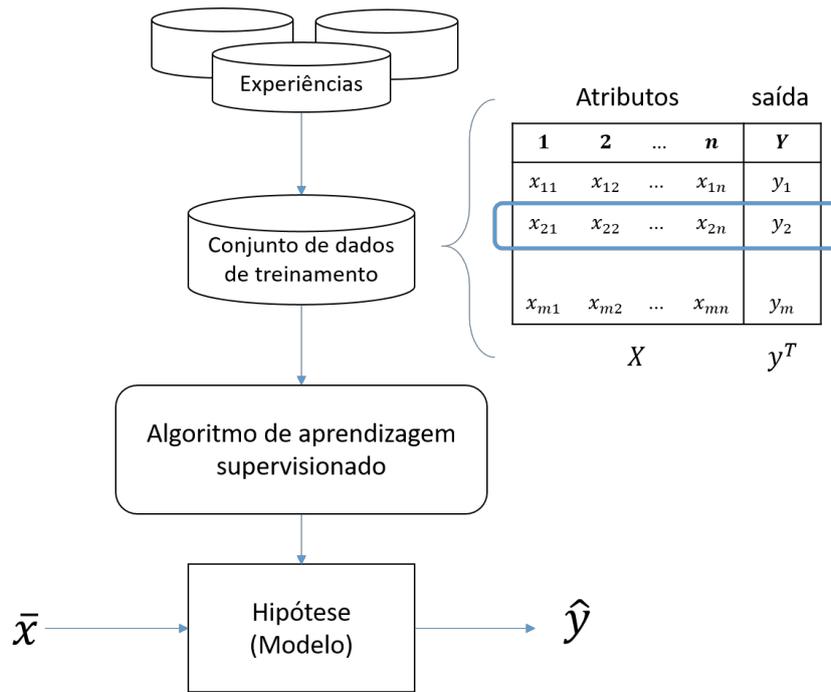


Figura 5. Processo de aprendizado de máquina supervisionado

A construção do conjunto de dados de treinamento é obtida através do processamento das experiências coletadas bem como da extração dos atributos de interesse. A representação desse conjunto de dados é feita geralmente de forma matricial. Na Figura 5, o conjunto de dados de treinamento equivale a m amostras $\{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_m, y_m)\}$. Este conjunto de dados é representado pela matrix X , onde cada linha \bar{x}_i ($i = 1, 2, \dots, m$) identifica os n atributos/características de interesse de cada experiência (variáveis de entrada) e seu valor de saída y_i .

Os algoritmos de regressão linear procuram obter o melhor modelo $h(\bar{x})$ que se ajuste ao conjunto de dados (entrada, saída) de treinamento. Este modelo é representado por uma função linear que descreve a relação entre uma ou mais variáveis de entrada e uma variável de saída estimada, tal como ilustra a equação 3.1:

$$h(\bar{x}_i) = \theta_0 + \theta_1 x_{i,1} + \theta_2 x_{i,2} + \dots + \theta_n x_{i,n} = \hat{y}_i, \quad (3.1)$$

onde θ_i representam os parâmetros (ou coeficientes) que determinam o mapeamento $h(\bar{x}) \rightarrow \hat{y}$. O objetivo do algoritmo é escolher o conjunto de

parâmetros $\{\theta_i, i = 1, 2, \dots, n\}$ de tal forma que $h(\bar{x})$ se aproxime o máximo possível de y conforme o conjunto de dados de treinamento. Para obter os coeficientes θ_i , o método de estimação por mínimos quadrados minimiza a soma do quadrado dos resíduos ($r_i = y_i - \hat{y}_i$), sendo estes definidos como a diferença entre os valores observados de uma amostra y_i e seus respectivos valores esperados $\hat{y}_i = h(\bar{x}_i)$. A equação 3.2 ilustra a função de custo a ser minimizada.

$$J(\theta) = \sum_{i=1}^m [y_i - h(\bar{x}_i)]^2 \quad (3.2)$$

onde m representa o número de pontos de dados utilizados na regressão linear.

Tal como se mencionou previamente, os valores dos parâmetros θ_i devem ser escolhidos com o objetivo de minimizar a função de custo $J(\theta)$. Este procedimento pode ser realizado através da solução de um sistema de equações normais [60] ou com o uso de algoritmos de busca tais como o método do gradiente (*gradient descente*)

3.2. Aprendizagem por reforço

A aprendizagem por reforço (*Reinforcement Learning*, RL) é o problema encontrado por um agente autônomo que deve aprender como agir em um ambiente dinâmico através de iterações de tentativa e erro [61]. Este tipo de aprendizagem é modelado como um processo de decisão de Markov (*Markov decision process*, MDP) onde o agente ligado ao ambiente, percebe/recebe como entrada uma indicação do **estado atual** \mathbf{s} do ambiente. Em seguida, o agente escolhe uma **ação** \mathbf{a} que gera uma saída. A ação tomada muda o estado do ambiente, e como resultado dessa mudança o agente recebe o valor referente à transição de estados a través de um **signal de reforço** \mathbf{r} , tal como ilustra a Figura 6. O objetivo do agente será então, achar a política (função) $\boldsymbol{\pi}(\mathbf{s})$ que mapeie os estados para ações que maximizem o valor final da soma dos reforços recebidos em um intervalo de tempo determinado.

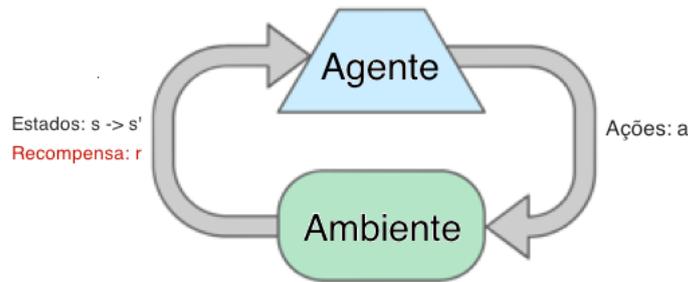


Figura 6. Agente com aprendizagem por reforço

Ao contrário dos métodos de aprendizado supervisionado, no aprendizado por reforço não existem pares “entrada/saída”, para serem utilizados no treinamento. Após tomar uma ação, o agente imediatamente recebe uma recompensa, mas não fica sabendo qual deveria ser a melhor ação para atingir o objetivo (ou maximizar o retorno a longo prazo). Ele precisa obter experiência dos possíveis estados, ações, transições e recompensas do sistema para atingir a melhor solução.

3.2.1. Exploração vs. aproveitamento

Uma das principais diferenças entre a aprendizagem supervisionada e a aprendizagem por reforço é que nesta última o agente deve explorar de forma explícita seu entorno, dado que não possui informação sobre as probabilidades de transição nem sobre as recompensas. Para isso, o agente deve definir uma taxa de exploração que permita inicialmente realizar ações de forma aleatória em cada estado, “explorando” as consequências sobre o objetivo definido e percebendo quais pares estado/ação maximizam as recompensas. Uma vez que a aprendizagem é feita, o agente diminuirá essa taxa de exploração para realizar em maior proporção as ações que produzem maiores retornos (fase de aproveitamento).

3.2.2. Processos de decisão de Markov: conceitos gerais

Os problemas relacionados com a aprendizagem por reforço são modelados como MDPs. Tal como ilustra a Figura 7, um processo de decisão de Markov (*Markov Decision Process*, MDP) é caracterizado por:

- Um conjunto de estados $s \in S$
- Um conjunto de ações $a \in A$

- Uma função de transição $T(s, a, s')$, que expressa a probabilidade de mudar de um estado s , para um estado s' realizando a ação a .
- Uma função de recompensa $R(s, a, s')$ que especifica a recompensa instantânea esperada em função do estado atual s e da ação a .

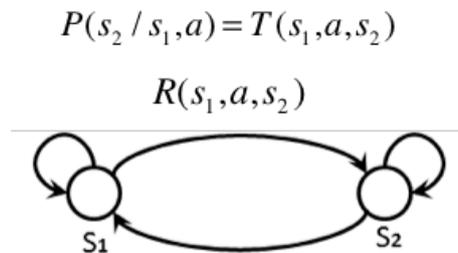


Figura 7. Processo de Markov

A Figura 8 ilustra como cada estado de um MDP pode ser visualizado como uma árvore de busca. Aqui, para cada estado s existe a possibilidade de realizar um conjunto de ações $a \in A$. Uma vez que o agente decide que ação realizar, este passa a um estado chamado de estado q (ou nó de probabilidade), no qual ainda não é conhecido o resultado de dita ação (próximo estado).

O modelo resultante é caracterizado como um MDP se as transições de estado são independentes de qualquer estado anterior ou das ações tomadas previamente pelo agente.

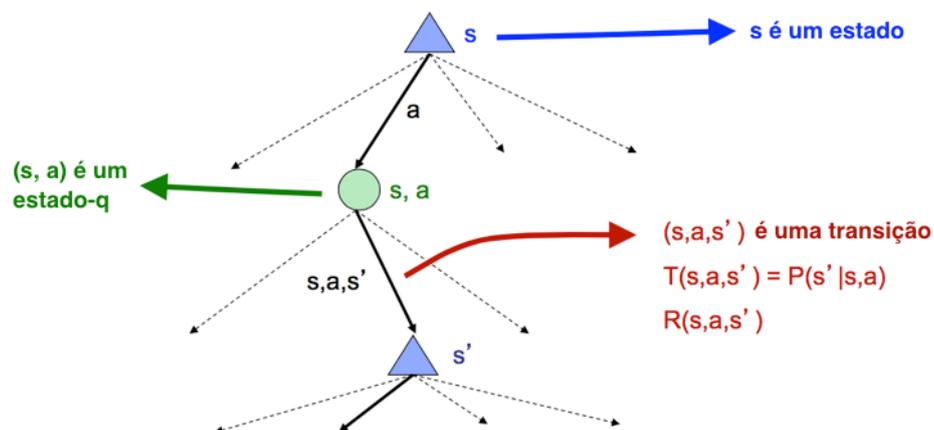


Figura 8. Processo de Markov (MDP) visto como uma árvore de busca

3.2.3.

Determinando as políticas ótimas a partir dos valores de $T(s, a, s')$ e $R(s, a, s')$

Uma política π é uma função que modela o comportamento do agente, mapeando estados em ações $\pi(s) \rightarrow a$. Uma política $\pi^*(s)$ é dita ser ótima

quando ela maximiza a recompensa cumulativa obtida pela execução de uma sequência de ações (R é também chamada de utilidade de uma sequência).

$$R_k = r_k + \gamma r_{k+1} + \gamma^2 r_{k+2} + \dots = \sum_{t=0}^{\infty} \gamma^t r_{k+t} \quad (3.3)$$

Onde r_k corresponde à recompensa no estado inicial k do agente e r_k denota a recompensa numérica instantânea obtida como resultado de realizar uma determinada ação no instante $t + k$. O γ é o fator de desconto que determina a importância relativa das futuras recompensas. Pequenos valores de $0 \leq \gamma \leq 1$ indicam um foco apenas nas recompensas instantâneas, enquanto que para valores perto de 1 serão valorizadas também as recompensas futuras.

Com isso, define-se o valor ótimo de um estado $V^*(s)$ como a utilidade futura esperada que pode ser obtida desde um estado quando se atua de forma ótima.

$$V^*(s) = \max_{\pi} \left(\sum_{t=0}^{\infty} \gamma^t r_{k+t} \right) \quad (3.4)$$

O valor da função $V^*(s)$ é único e pode ser obtido de forma recursiva a partir das equações de Bellman [61], [62].

Da Figura 8, observa-se que o valor ótimo no estado s é obtido escolhendo a ação que produz o melhor resultado em termos dos valores dos estados q (ou nós de probabilidade) ótimos, ou seja,

$$V^*(s) = \max_a Q^*(s, a), \quad (3.5)$$

onde a expressão para o valor ótimo de um *estado- q* pode ser escrita como a média sobre todos os possíveis valores obtidos nos estados futuros s' .

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s)] \quad (3.6)$$

A partir das equações (3.5) e (3.6) obtém-se a equação recursiva que define a utilidade do estado s .

$$V^*(s) = \max_a \left(\sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s)] \right) \quad (3.7)$$

Finalmente, a partir do valor ótimo do estado é possível definir a política ótima para o dito estado, como segue:

$$\pi^*(s) = \underset{a \in S}{\operatorname{argmax}} \left(\sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s)] \right), \forall s \in S \quad (3.8)$$

O sistema de equações caracteriza os valores ótimos para os estados, mas, não específica como computá-los. Para isso, são utilizados algoritmos e mecanismos computacionais que tomam as equações de Bellman descritas e as transformam em métodos para a obtenção dos valores ótimos esperados em cada estado. Dentre os principais algoritmos utilizados na literatura podem ser mencionados a iteração do valor e a iteração de políticas [61].

3.2.4. Métodos de aprendizado

Na seção anterior foi apresentada de forma sucinta a metodologia para obter as políticas ótimas para um MDP assumindo que se conta com o modelo do sistema, composto pelo conjunto de estados S , o conjunto de ações A , a função de transição de probabilidade $T(s, a, s')$ e a função de recompensa $R(s, a, s')$. O grande desafio nos problemas de aprendizado por reforço é que o agente não conta com o modelo para resolver o MDP, portanto, o agente deverá interagir com ambiente de forma direta para obter informações suficientes que permitam (através de algoritmos apropriados) a obtenção das políticas ótimas $\pi^*(s)$.

Para isso, na literatura encontram-se os métodos seguintes:

- Baseados em modelo (*Model-based*): O agente aprende um modelo empírico a partir de experiências (amostras) para depois resolver o MDP.
- Livres de Modelo (*Model-free*): O agente aprende as políticas ótimas sem a necessidade de aprender o modelo do sistema.

3.2.5. Algoritmo Q-Learning (livre de modelo)

No algoritmo *Q-learning*, o agente tenta aprender as políticas ótimas a partir do histórico de iterações/experiências com o meio, representado por uma sequência estado-ação-recompensa, ou seja,

$$(s_0, a_0, r_1, s_1, a_1, r_2, s_3, a_3, r_4 \dots) \quad (3.9)$$

o agente no estado s_0 realizou a ação a_0 recebendo uma recompensa r_1 e evoluindo para o estado s_1 . Em seguida, ele realizou a ação a_1 , recebeu a recompensa r_2 e terminou no estado s_2 ; e assim por diante. Este histórico de iterações é modelado pela sequência (s, a, s', r) para representar cada transição entre estados.

O *Q-learning* utiliza diferenças temporais para estimar os valores de $Q(s, a)$ cada vez que recebe uma amostra (transição s, a, r, s'), de tal forma que o agente consiga manter uma tabela atualizada que representa as estimativas do recompensa esperada ao realizar a ação a no estado s .

Matematicamente pode ser visto da seguinte forma: cada experiência (s, a, r, s') fornece uma amostra que sugere o valor futuro de $Q(s, a)$ composto pela atual recompensa mais o futuro valor estimado (com desconto).

$$Q(s, a) \approx r + \gamma V^*(s') = r + \gamma \max_{a'} Q(s', a') \quad (3.10)$$

em seguida, a tabela $Q(s, a)$ é atualizada incorporando o retorno recebido (amostra) por de diferenças temporais:

$$Q(s, a) \leftarrow Q(s, a) + \lambda \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \text{ ou} \quad (3.11)$$

$$Q(s, a) \leftarrow (1 - \lambda)Q(s, a) + \lambda[r + \gamma Q(s', a')], \quad (3.12)$$

onde γ representa o fator de desconto e λ a taxa de aprendizado. O fator de aprendizagem pondera a influência de uma atualização recente no valor de $Q(s, a)$ e o fator de desconto determina a importância das recompensas futuras.

O pseudocódigo do algoritmo *Q-learning* é apresentado na sequência.

Algoritmo 1. Q-Learning

1. Inicializar a tabela para armazenar os valores-q
 $q[i, j] = 0$ para $1 \leq i \leq N$ e $1 \leq j \leq J$
 Sendo N o número de estados possíveis e J o número de ações por estado.
 2. Observar o estado atual S
 3. Selecionar uma ação a a partir do estado atual s e conforme a estratégia de exploração/aproveitamento escolhida.
 4. Realizar a ação escolhida e deixar o sistema evoluir ao próximo estado s' .
 5. Observar o sinal de reforço (recompensa) r'
 6. Atualizar a tabela $Q(s, a)$ através de diferenças temporais.

$$Q(s, a) \leftarrow Q(s, a) + \lambda \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$
 onde γ representa o fator de desconto e λ a taxa de aprendizado.
 7. $s \leftarrow s'$
 8. Repetir o processo desde o ponto 2.
-

No próximo capítulo, apresenta-se a arquitetura cognitiva proposta para apoiar os processos de auto-organização em redes móveis, bem como a estrutura de simulação desenvolvida para implementar e validar as técnicas cognitivas propostas nesta tese.

4

Arquitetura cognitiva e cooperativa para redes móveis auto-organizáveis

Neste capítulo é descrito o processo de desenvolvimento da arquitetura para a integração de processos cognitivos que apoiem as funções de auto-organização nas redes de acesso móvel. Inicialmente, é apresentada uma visão geral do processo de definição da arquitetura, para em seguida detalhar os princípios que foram adotados para seu desenvolvimento. Posteriormente, apresenta-se sua estrutura e seus componentes funcionais, ressaltando como é feita sua integração na rede de acesso. Finalmente, apresenta-se a estratégia de modelagem e simulação utilizada neste trabalho para validar sua aplicabilidade na resolução de problemas no âmbito de redes móveis auto-organizáveis.

4.1. Descrição geral

A inclusão de técnicas auto-adaptativas nas redes de acesso sem fio tem se tornado um fator chave para lidar com o aumento na complexidade e nos custos de operação/manutenção da rede [2]. Este tipo de abordagem visa desenvolver mecanismos que permitam à rede ter comportamentos autônomos para se adaptar às mudanças no ambiente, otimizando o desempenho do sistema com pouca ou nenhuma intervenção humana. De forma complementar, o conceito de gerenciamento cognitivo de redes é abordado na literatura como uma evolução dos processos de gerenciamento automático, onde a rede e o sistema de gerenciamento são enriquecidos com capacidades de autoaprendizagem inspiradas nas teorias da cognição humana [63]. Os primeiros conceitos nesta área de pesquisa em tecnologias sem fio, surgiram do trabalho pioneiro sobre rádio cognitivo de J. Mitola e Maguire [6]. Posteriormente, estes conceitos foram estendidos para a totalidade da rede por R. Thomas, DaSilva e MacKenzie [9] que definem conceito de redes cognitivas como segue:

“Uma rede cognitiva é aquela que integra um processo cognitivo no qual o sistema percebe seu estado atual para então planejar, decidir e atuar sobre essas condições. Esta rede é capaz de aprender a partir de adaptações realizadas no passado e usar este conhecimento para tomar decisões futuras, enquanto leva em consideração objetivos de transmissão fim-a-fim” (tradução do autor)¹.

Nesta tese, foi adotada essa definição no desenvolvimento de uma arquitetura cognitiva e cooperativa focada nos processos de auto-organização na rede de acesso móvel, incluindo todas as camadas da pilha de protocolos rádio e levando em consideração objetivos de desempenho fim-a-fim.

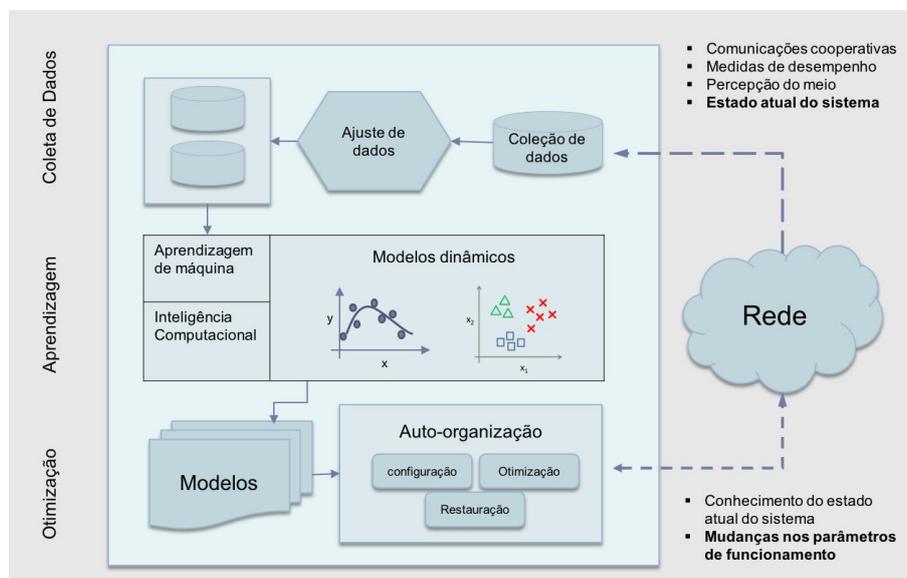


Figura 9. Fluxo de informação em um dispositivo de acesso com capacidades cognitivas e de auto-organização

O objetivo principal desta arquitetura é dotar diversos dispositivos da rede de acesso com capacidades de aprendizado a partir de técnicas de inteligência computacional e aprendizado de máquina. A Figura 9 ilustra o fluxo de informação que permite aos dispositivos da rede de acesso construir dinamicamente os modelos ou elementos de apoio à decisão sobre quais as funcionalidades SON são implementadas. Este fluxo de informação se baseia na construção de um *loop de feedback* no qual as interações realizadas previamente com o ambiente guiam as ações futuras. Para tal fim, o dispositivo realiza uma

¹ “A cognitive network has a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals” [9].

coleta de dados referentes ao estado atual do sistema através de medidas de desempenho locais e comunicações cooperativas. Esta informação é posteriormente processada por técnicas de aprendizagem, associando dinamicamente indicadores e métricas de desempenho com os parâmetros de configuração/operação da rede (Figura 10), resultando nos modelos ou elementos que apoiam os processos de auto-organização.

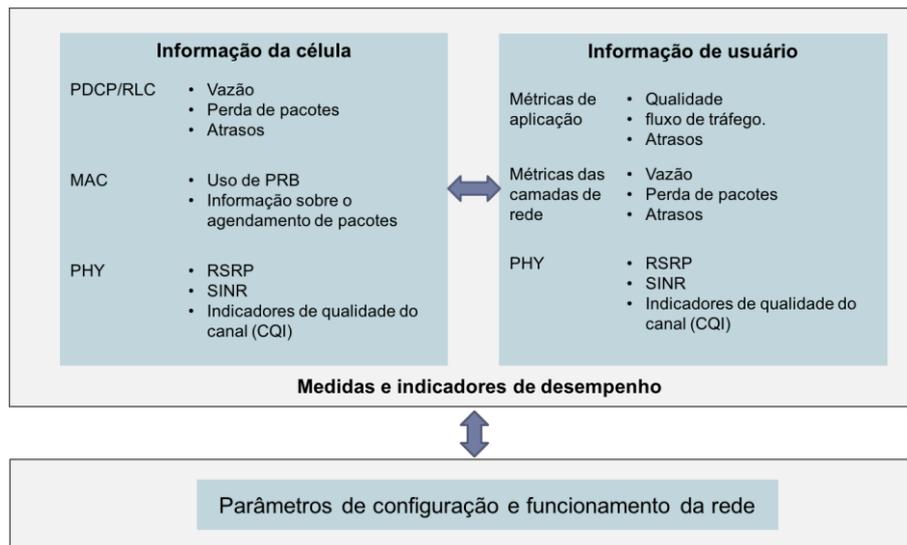


Figura 10. Fontes de informação para serem utilizadas pela técnica de aprendizagem

4.2. Princípios de desenvolvimento

A seguir, apresentam-se os principais conceitos de desenvolvimento considerados neste trabalho para abordar os desafios da integração de funcionalidades cognitivas nas arquiteturas de acesso rádio móvel.

4.2.1. Soluções distribuídas

Um sistema de controle distribuído é caracterizado por ter suas funções principais controladas localmente em diferentes pontos, ao contrário de ter um único ponto de controle central para todo o sistema. Esta característica é essencial para superar problemas de escalabilidade devido ao incremento de elementos na rede de acesso rádio.

4.2.2. Modularidade

Do ponto de vista do *design*, a modularidade é uma estratégia que subdivide um sistema em partes menores (módulos) que podem ser criados de forma independente. Os módulos representam uma separação de interesses, facilitando aos desenvolvedores de rede a redução da inerente complexidade de um sistema cognitivo. Além disso, o desenvolvimento modular oferece outros benefícios em termos de flexibilidade tais como a exclusão, a modificação ou o acréscimo de novas soluções no tempo.

4.2.3. Self-Awareness

Neste documento, o termo de *self-awareness* é utilizado para indicar a capacidade da rede de perceber seu estado atual de funcionamento, interpretá-lo, decidir e atuar sobre essas condições para atingir objetivos de desempenho. Tomando em consideração os requisitos de escalabilidade e o princípio de soluções distribuídas, argumenta-se que as capacidades *self-awareness* devem ser estendidas aos elementos de acesso na rede rádio. Por isso, na arquitetura aqui desenvolvida, *self-awareness* representa a capacidade das entidades da rede de acesso de perceber seu estado atual de operação, assim como também da percepção do impacto causado pela presença de outras entidades vizinhas.

4.2.4. Capacidades de aprendizagem

Nesta tese, adotamos a definição de aprendizagem relacionada com o uso de algoritmos de inteligência computacional utilizada por Thathachar [64]:

“Se diz que uma máquina ou sistema aprende se este melhora o seu desempenho através da experiência adquirida ao longo de um período de tempo sem informações completas sobre o ambiente em que opera” (tradução do autor)².

² “A machine or system is said to learn if it improves its performance through experience gained over a period of time without complete information about the environment in which it operates.” [64].

Assim, além da capacidade de prever (ou antecipar) possíveis eventos na rede, é esperado que os dispositivos com capacidades cognitivas se tornem cientes do resultado das suas ações, permitindo ao sistema otimizar seu desempenho no tempo, mesmo em presença de um ambiente complexo e evolutivo.

4.2.5. Desenvolvimentos multicamada

Tradicionalmente, as pilhas de protocolos de comunicação são compostas por camadas conceituais bem definidas, isolando suas funcionalidades em estruturas fechadas e mantendo uma interação entre elas através de interfaces padronizadas. O desenvolvimento multicamada visa quebrar esse encapsulamento na procura de uma otimização mais eficaz da qualidade da comunicação fim-a-fim. Esta abordagem multicamada possibilita unificar dados de diferentes fontes de informação e explorar as relações entre vários indicadores de desempenho e parâmetros de configuração de rede.

4.2.6. Capacidades de compartilhamento de informação

Dado que as funcionalidades SON são modeladas como estruturas adaptativas baseadas em interações locais das entidades do sistema, é necessário estabelecer mecanismos e interfaces que permitam o compartilhamento de informações de estado e de desempenho que sustentem as funcionalidades de aprendizagem e de otimização. Assim, os elementos pertencentes à rede cognitiva terão a possibilidade de trabalhar cooperativamente para aprender como responder de forma apropriada às mudanças do ambiente conforme os objetivos definidos pelo operador.

4.3. Arquitetura funcional

Nesta seção, é apresentado o processo de desenvolvimento da arquitetura cognitiva e cooperativa para redes móveis auto-organizáveis, que denominamos CCSON (acrônimo em inglês para *cognitive and cooperative for self-organizing mobile networks*). Projetamos esta arquitetura com base nos conceitos

relacionados com o gerenciamento autônomo de redes [34], redes cognitivas [9] e computação móvel na borda [65]. Especificamente, adotamos os princípios propostos pela IBM no seu modelo de computação autônoma MAPE-K (*The Monitor-Analyze-Plan-Execute over a shared Knowledge*) [10].

CCSON é uma arquitetura distribuída baseada em agentes de *software* associados com os elementos da rede de acesso. Nela, cada agente é visto como um processo computacional capaz de realizar múltiplas funções de forma autônoma. Especificamente, cada agente cognitivo deve cumprir com as seguintes funções: 1) estar ciente do estado de operação atual do dispositivo de acesso, 2) comunicar e trabalhar de forma cooperativa com outros agentes para 3) tomar decisões e mudar parâmetros de rede e 4) aprender das experiências passadas para melhorar o desempenho do sistema ao longo do tempo.

Como se observa na Figura 11, as funcionalidades dos agentes foram desenvolvidas através de elementos independentes que representam cada uma das funcionalidades necessárias para sua operação, sendo eles: elementos de operação (EO), elementos de percepção (EP) e módulos de auto-organização (SOM, por suas siglas em inglês: *self-organizing module*) que por sua vez foram agrupados em três camadas funcionais: percepção, decisão/aprendizagem e operação. A descrição dessas camadas é apresentada nas subseções a seguir.

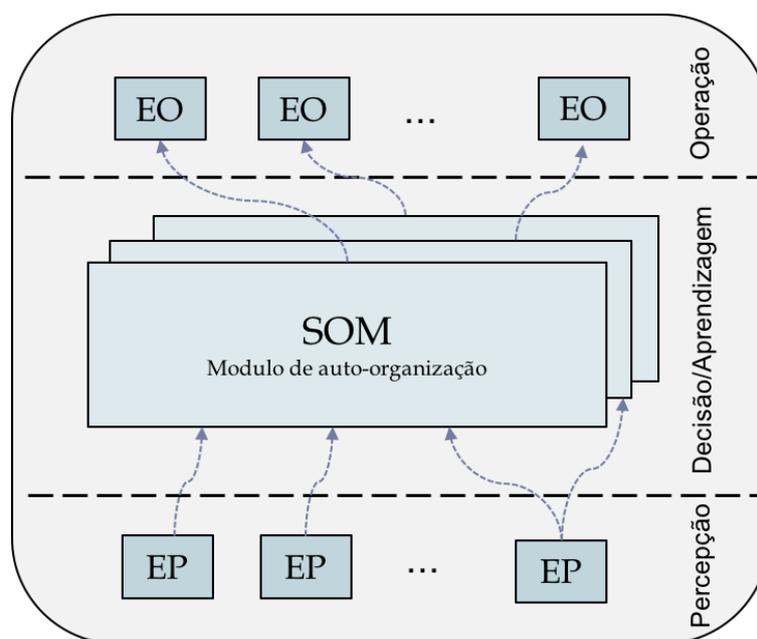


Figura 11. Arquitetura funcional do agente cognitivo

4.3.1. Camada de percepção

As funções incluídas na camada de percepção são as responsáveis de obter, processar e armazenar temporariamente as informações obtidas dos elementos da rede de acesso rádio. Entre os principais dados de interesse estão as métricas de desempenho e os parâmetros de configuração que estejam sendo utilizados no momento. Esta camada incorpora também as funcionalidades encarregadas da comunicação (recepção) e cooperação com agentes vizinhos. A informação nessa camada é gerenciada por entidades individuais chamadas de elementos de percepção (EP), os quais poderão ser utilizados simultaneamente por vários módulos de auto-organização.

4.3.2. Camada de decisão/aprendizagem

A camada de decisão/aprendizagem foi projetada para agrupar as funcionalidades de auto-organização, que são implementadas através de módulos independentes chamados de módulos de auto-organização (SOM). Esta abordagem modular pretende facilitar a modificação e integração de novas soluções, além de permitir abordar o desenvolvimento de cada função de auto-organização (SON) através do paradigma de desenvolvimento de *software* denominado “separação de interesses” [66]. Desta forma, procura-se encapsular em cada módulo a informação e os mecanismos de decisão de forma independente, permitindo a utilização de algoritmos de otimização ou aprendizado automático apropriados para cada aplicação SON.

Internamente, como mostra a Figura 12, cada módulo possui uma estrutura de funções próprias relacionadas com o processo de decisão/aprendizagem, sendo estas definidas pela formulação específica da aplicação a ser implementada. Estas funções representam a capacidade do agente de integrar processos de comunicação/cooperação com outros agentes, de tomada de decisão (otimização), e de aprendizagem (construção de modelos ou elementos de apoio a decisão).

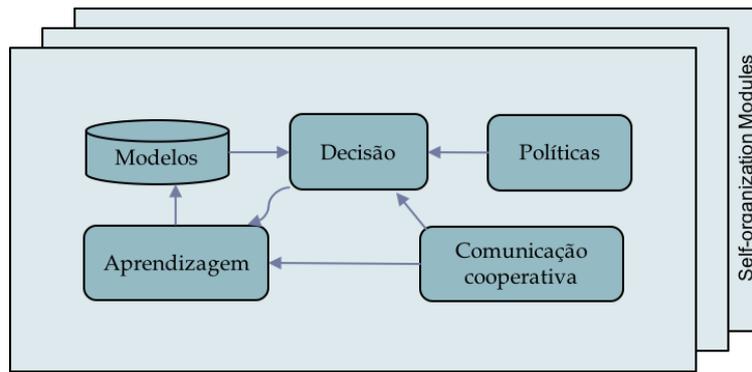


Figura 12. Estrutura interna de um módulo de auto-organização cognitivo

A comunicação exerce um importante papel dentro da arquitetura CCSON, permitindo a coordenação de agentes dentro do sistema. Para isso, cada módulo SOM incorpora/encapsula suas próprias estratégias de comunicação, permitindo a criação de grupos cognitivos entre agentes que compartilham as mesmas funcionalidades (Figura 13). Estes grupos cognitivos permitem a realização de troca de informações necessárias para atingir os objetivos de aprendizagem e desempenho desejados.

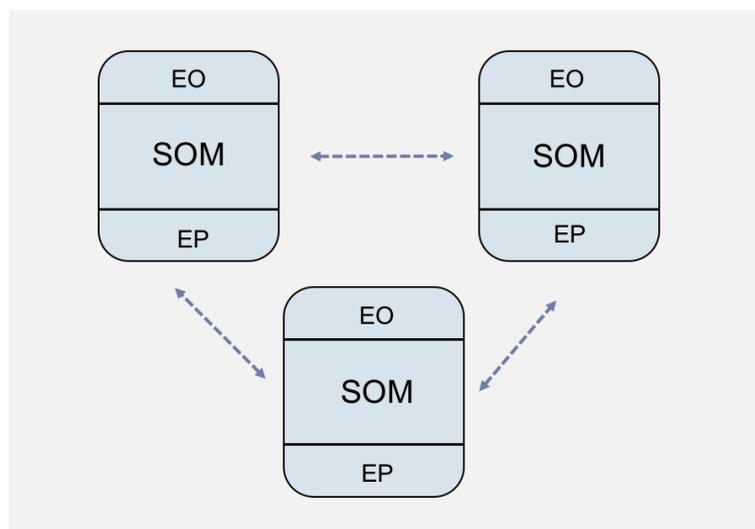


Figura 13. Formação de grupos cognitivos

4.3.3. Camada de operação

A camada de operação é a responsável por realizar os ajustes e reconfigurações dos parâmetros de rede nos dispositivos de acesso como resultado das decisões tomadas pelos módulos de auto-organização na camada de aprendizado/otimização. Esta camada também incorpora funcionalidades

encarregadas da comunicação (transmissão) e cooperação com agentes vizinhos. A informação nesta camada é gerenciada por entidades individuais chamadas de elementos de operação. No caso em que diferentes SOM realizem mudanças sobre um mesmo parâmetro de operação de rede, o elemento de operação responsável por dito parâmetro deverá incluir mecanismos de coordenação [67] e resolução de conflitos conforme as políticas definidas pelo operador móvel.

4.4. Integração na rede de acesso rádio móvel.

O conceito de computação móvel na borda [65] propõe a integração de soluções de *software* dentro da rede de acesso rádio e na proximidade dos usuários móveis, reduzindo a latência e garantindo uma operação de rede altamente eficiente. De forma similar, a arquitetura CCSON, integra os agentes cognitivos na proximidade dos elementos de acesso rádio. Para isso, foi utilizada uma interface multicamada para facilitar o intercâmbio de informações entre as camadas de percepção/operação do agente cognitivo e o dispositivo de acesso. A Figura 14 ilustra um exemplo deste conceito aplicado às arquiteturas rádio nas tecnologias *Long-Term Evolution* LTE/LTE-A. Usando este conceito, foi possível limitar o impacto sobre os dispositivos de usuário adicionando apenas as funções cognitivas na infraestrutura da operadora. No entanto, os agentes podem obter informações sobre o desempenho da comunicação rádio fim-a-fim na interface rádio (eNB-UEs) além de métricas relativas aos terminais de usuário graças aos mecanismos de gerenciamento existentes nas recentes tecnologias móveis.

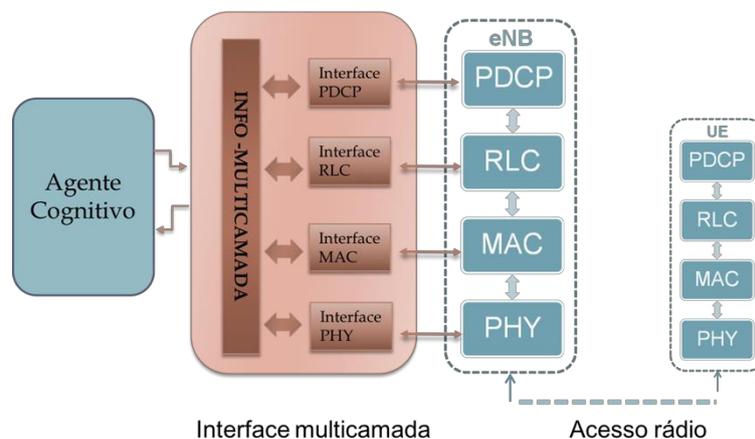


Figura 14. Interface multicamada

4.5. Modelagem e simulação

Nesta tese testamos os conceitos chave da arquitetura CCSON através da implementação de mecanismos cognitivos e auto-organizáveis de balanceamento de carga para redes móveis LTE heterogêneas (apresentados nos capítulos 5 e 6). Para isso, realizamos uma análise baseada em simulação através da ferramenta de código aberto NS-3 (network simulator 3) [29]. O NS-3 é um simulador de eventos discretos orientado ao estudo de redes de pacotes, escrito na linguagem C++ e distribuído com licença GNU GPLv2. Este tipo de simulador orientado a eventos, modela o funcionamento do sistema como uma sequência discreta de eventos no tempo. Assim, cada evento ocorre em um determinado instante no tempo e marca uma mudança de estado no sistema [68].

Especificamente, para simular a rede de acesso LTE, utilizamos o módulo LENA [30] desenvolvido pelo Centro Tecnológico de Telecomunicações de Catalunha. Este módulo está atualmente integrado nas bibliotecas do simulador NS-3, destacando-se pelo nível de detalhe do modelo, emulando as especificações técnicas definidas nos padrões do 3GPP. Funcionalmente, o módulo LENA é formado por dois componentes principais [69]:

- O modelo LTE: este modelo inclui a pilha do protocolo rádio de LTE (RRC, PDCP, RLC, MAC, PHY). Estas instancias residem dentro dos nós de usuário (UE) e as estações rádio-base (eNB).
- O modelo de núcleo de rede - (*Evolved Packet Core, EPC*): Estes modelos incluem a implementação das interfaces, protocolos e entidades principais do núcleo de rede. Estas entidades e protocolos residem dentro dos nós *serving gateway*, *packet data network gateway*, *mobile management entity* e parcialmente dentro dos nós eNB.

A Figura 15.a ilustra a visão geral das entidades modeladas pelo NS-3 enquanto na Figura 15.b observa-se a pilha de protocolos implementada no plano de usuário do núcleo de rede e interface do ar.

Durante a realização do trabalho, adicionamos ao simulador NS-3 as classes e métodos necessários para integrar a estratégia de expansão virtual de cobertura (*Cell Range Expansion, CRE*), utilizada como base no desenvolvimento das técnicas de balanceamento de carga. Além disso, realizamos as modificações

necessárias para calcular os valores das medidas de desempenho da rede de acesso referentes a utilização dos blocos de recurso de cada eNB.

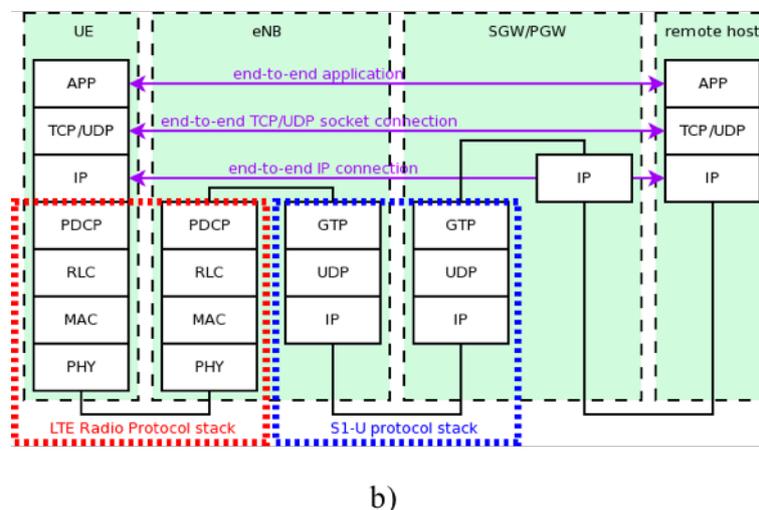
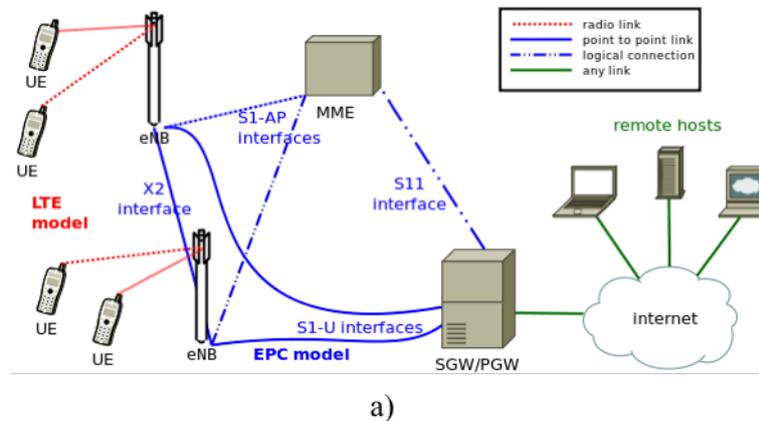


Figura 15. Visão geral do módulo LTE-EPC implementado no simulador NS-3: a) entidades modeladas, b) pilha de protocolos no plano de dados do núcleo de rede e a interface aérea [69]

A Figura 16 ilustra os programas envolvidos na modelagem e simulação da arquitetura CCSON. Tal como mencionado anteriormente, os cenários da rede de acesso LTE foram executados no simulador NS-3, enquanto a estrutura funcional e os processos matemáticos relacionados com os agentes cognitivos foram implementados na ferramenta Matlab (segundo os detalhes conceituais definidos na seção 4.3). Conjuntamente, utilizaram-se programas de linha de comando Unix (*shell scripts*) como interface de comunicação entre as ferramentas Matlab e NS-3.

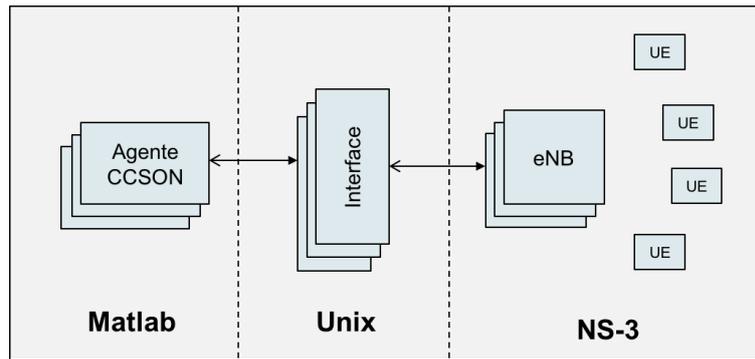


Figura 16. Arquitetura do software desenvolvido para a modelagem e simulação da arquitetura CCSON

Na ferramenta NS-3, a simulação dos eventos da rede de acesso foi configurada para ter uma duração de n períodos de observação com duração T . Durante cada período de observação, os elementos de acesso obtêm as estatísticas relacionadas com o funcionamento do sistema através de métricas de desempenho e parâmetros de operação. A seguir, esta informação é comunicada aos agentes cognitivos (no Matlab) que processam dita informação através de suas funções cognitivas.

Tal como mostra a Figura 17, todo o processo de simulação é controlado por um fluxo de trabalho desenvolvido em Matlab, que coordena a troca de informações e a interação entre os programas envolvidos.

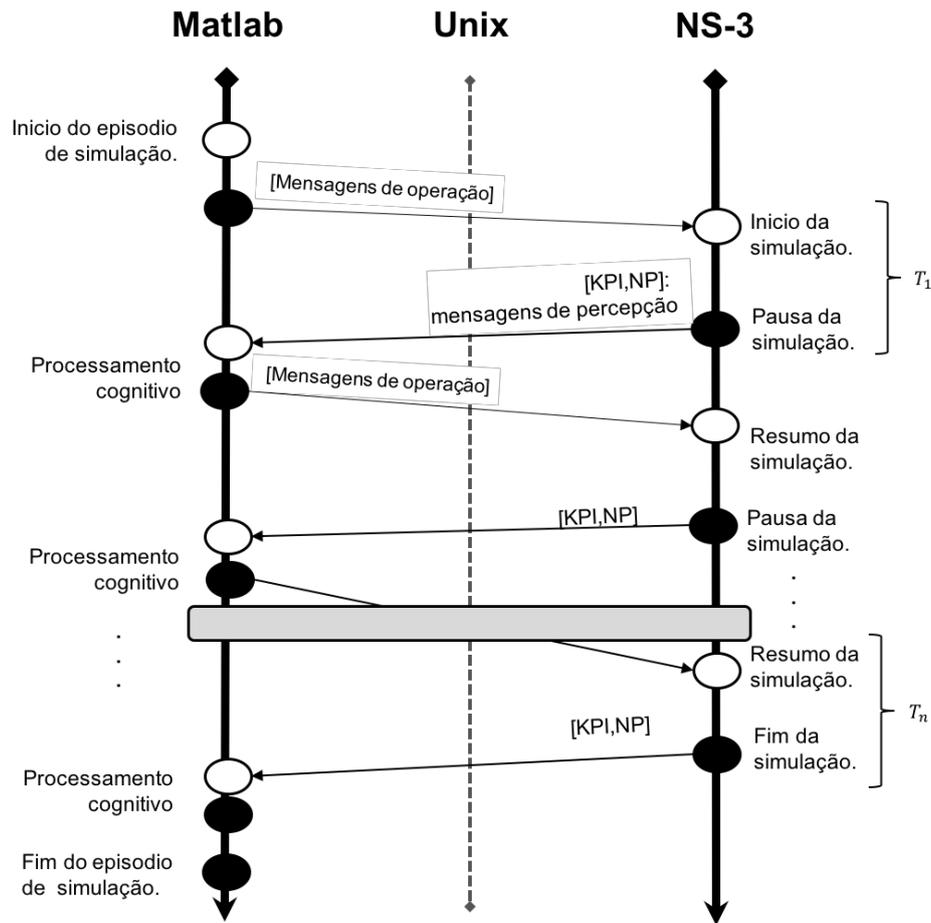


Figura 17. Iteração entre as ferramentas Matlab e NS-3

No capítulo seguinte, apresenta-se o desenvolvimento de técnicas SON para o balanceamento de carga em redes móveis e heterogêneas, apoiadas em mecanismos de aprendizagem de máquina.

5

Técnicas cognitivas de balanceamento de carga em redes móveis heterogêneas.

No decorrer deste capítulo abordamos o problema do balanceamento de carga em rede móveis entre macro e pico células [28]. Este problema é tratado como um estudo do caso para mostrar como a arquitetura CCSON apresentada no capítulo anterior pode ser utilizada para melhorar o desempenho da rede de acesso rádio. Na seção 5.1 apresenta-se em detalhe o problema tratado e a técnica empregada para o desenvolvimento da funcionalidade SON de balanceamento de carga e na seção 5.2 descrevem-se as características do cenário simulado.

Especificamente, neste capítulo desenvolvemos duas técnicas cognitivas SON utilizando diferentes mecanismos de inteligência computacional. A primeira técnica, apresentada na seção 5.3, utiliza uma abordagem utilizando um algoritmo de aprendizagem supervisionado, enquanto a segunda técnica apresentada na seção 5.4, utiliza outra abordagem baseada em uma metodologia de aprendizagem por reforço.

Finalmente, a seção 5.5 conclui o capítulo com as observações e conclusões obtidas.

5.1.

Balanceamento de carga em redes móveis heterogêneas: seleção de célula e expansão virtual de cobertura

O uso de redes heterogêneas através da adição de uma camada de estações base de baixa potência é útil em casos de otimização de cobertura e capacidade, especialmente em lugares com distribuição de tráfego não uniforme, com necessidade de cobrir zonas de sombra ou melhorar as comunicações na borda da célula [70]. No entanto, devido às altas potências de transmissão das macro-eNBs (MeNB), em alguns casos não é possível trasladar a suficiente quantidade de UE da MeNB para as pico-eNBs (PeNB). Nos métodos tradicionais de seleção de célula, os UE comparam as potências dos sinais de referência recebidas

(*Reference Signal Received Power, RSRP*), escolhendo desta forma a estação base da qual é percebido o maior nível.

Para solucionar este problema pode-se utilizar uma técnica conhecida como “*cell range expansion*” (CRE) [12], que permite aumentar a cobertura das PeNB sem necessidade de incrementar suas potências de transmissão. Isso é conseguido ao adicionar um valor de *CRE-offset* (*CREO*) ao valor do RSRP durante o processo de seleção de célula nos terminais (Figura 18). Dessa forma, os UEs não vão se conectar com a célula com maior *RSRP*, senão com aquela que tiver maior ($RSRP + CREO$). Através da implementação do CRE é possível associar um maior número de UE com as PeNB incrementando o balanceamento de carga. No entanto, este benefício vem com o custo associado ao incremento dos níveis de interferência para os usuários agora localizados na borda das PeNB.

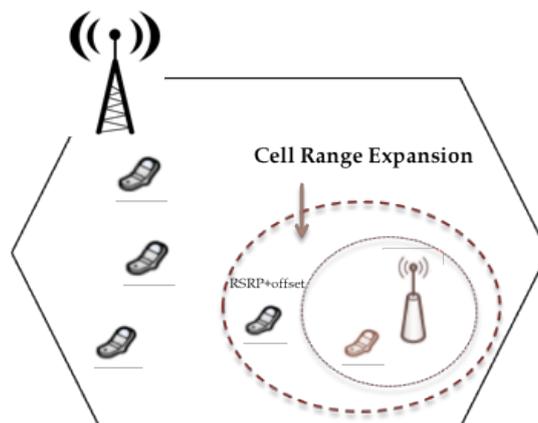


Figura 18. Técnica "cell range expansion"

Objetivo de aprendizagem: Especificamente, nas técnicas desenvolvidas neste capítulo, procuramos fazer com que o sistema aprenda a modificar dinamicamente o valor de *CREO* utilizado pela PeNB conforme as condições de tráfego da célula. Desta forma, a pico-célula incrementará sua cobertura contribuindo com o balanceamento de carga e a melhora no desempenho da célula.

5.2.

Caso de estudo e cenário de simulação

Neste trabalho foi considerado um cenário LTE heterogêneo com 7 sites e distâncias entre eles de 500 metros. Cada *site* é composto por três setores hexagonais e uma pico célula omnidirecional na borda da célula tal como se

ilustra na Figura 19. No cenário implementado, o centro da célula representa a área de interesse enquanto o resto de eNBs geram interferência aos usuários avaliados.

No cenário implementado, considerou-se o envio de mensagens entre MeNB e PeNB através da interface X2 da tecnologia LTE [71]. Desta forma, os elementos de acesso conseguem trocar informações de desempenho e do estado de operação, sustentando a criação de grupos cognitivos/cooperativos (Seção 4.3).

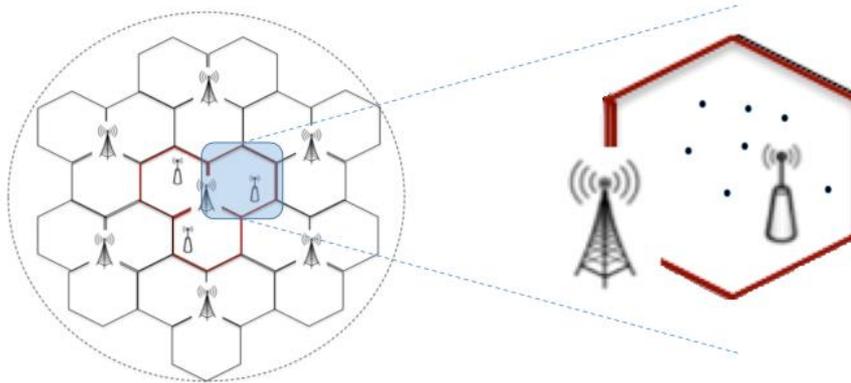


Figura 19. Cenário de estudo e simulação

A Tabela 1 apresenta os principais parâmetros e modelos de propagação utilizados na rede de acesso (*Evolved Universal Terrestrial Radio Access Network, E-UTRAN*). Estes valores estão baseados nos parâmetros definidos pela 3GPP para a simulação de sistemas heterogêneos, especificamente para o modelo 1 (*Urban Macro cell + outdoor Hotzone cells*, tabela A.2.1.2-3 no documento 3GPP-TR 36.814 [72]).

Tabela 1. Parâmetros de simulação para o E-UTRAN (Fonte: [72])

Parâmetro	Valor
Frequência da portadora	2 Ghz
Largura de banda	5 Mhz
Distância entre as MeNB (<i>ISD</i>)	500 m
Potência de transmissão MeNB	46 dBm
Potência de transmissão PeNB	30 dBm
Algoritmo de escalonamento de pacotes	<i>Proportional Fair</i>
Perdas de propagação - MeNB	$128.1 + 37.6 \cdot \log_{10} R$
Perdas de propagação - PeNB	$140.7 + 36.7 \cdot \log_{10} R$
Perdas de penetração	20 dB

A seguir, apresentam-se as técnicas cognitivas desenvolvidas neste capítulo para a realização do balanceamento de carga com a ajuda do mecanismo de expansão de cobertura CRE.

5.3.

Técnica 1: balanceamento de carga cognitivo através de uma abordagem de aprendizagem supervisionado

No desenvolvimento desta técnica foi adotada uma abordagem de aprendizado automático em funcionalidades *SON* relacionadas com o balanceamento de carga e a técnica *CRE* [33]. O objetivo é brindar à rede a capacidade de aprender a modificar dinamicamente o *CRE offset (CREO)* utilizado pela PeNB, reduzindo o congestionamento na macro-célula e diminuindo a perda de pacotes rede de acesso rádio.

No diagrama de blocos da Figura 20 observa-se a metodologia utilizada. Todo o processo ocorre em duas etapas: a fase de aprendizagem (geração de experiências e treinamento) e, posteriormente, a fase de seleção do *CREO* (otimização).

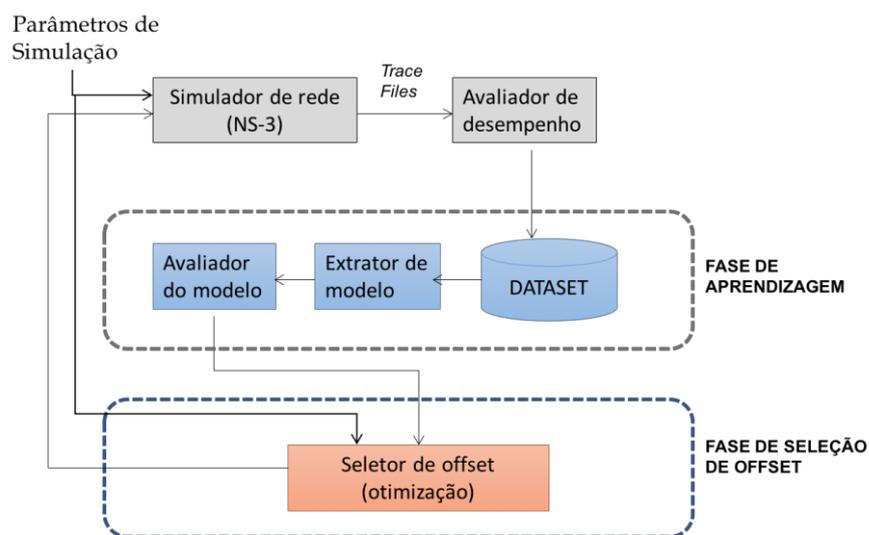


Figura 20. Diagrama de blocos do processo de aprendizado e seleção de CREO

A primeira etapa visa gerar e coletar um conjunto de dados de treinamento para a obtenção do modelo que relacionem as métricas de desempenho com o parâmetro *CREO* responsável pelo balanceamento de carga. A partir dessa

informação, a segunda etapa consiste na utilização dos modelos obtidos previamente para otimizar o desempenho da célula.

Apresentam-se a seguir os detalhes da metodologia utilizada.

5.3.1. Fase de aprendizagem

O objetivo da fase de aprendizagem é extrair (para um cenário em particular) dos dados de treinamento as relações entre os indicadores de desempenho e os valores do *CREO* responsáveis pela expansão de cobertura das PeNB. Especificamente, nesta etapa foi utilizada uma metodologia de aprendizado supervisionado [53] baseado em uma regressão polinomial multivariável, permitindo estimar a associação entre uma variável de saída(dependente) e múltiplas variáveis preditoras (ou variáveis explicativas) [73]. Esta etapa foi realizada seguindo o fluxo tradicional utilizado no aprendizado de máquina supervisionada: 1) obtenção dos dados de treinamento, 2) extração das características/atributos, 3) aplicação dos algoritmos de aprendizado para a extração do modelo e 4) avaliação dos resultados.

5.3.1.1. Obtenção dos dados de treinamento

Neste passo, foi construído o conjunto de dados de treinamento sobre o qual é feito o aprendizado. Este conjunto de dados deve ser construído a partir de observações que contenham uma quantidade suficiente de pontos de dados que envolvam toda a região de interesse. O desafio é obter a diversidade necessária nas medições requeridas a partir de uma rede operacional. Por tanto, é necessário testar uma ampla variedade de configurações para o *CREO* (incluindo valores que poderiam ser prejudiciais para o desempenho) em uma rede que é projetada para satisfazer os requisitos de QoS aos clientes. Uma das soluções é construir o conjunto de dados de treinamento a partir de simulações, e uma vez que as tendências sejam extraídas, aplicar o mapa corretivo necessário às simulações e medidas realizadas. Neste caso, a confiabilidade das previsões vai depender da precisão das simulações, bem como da qualidade das medições.

O conjunto de treinamento foi gerado usando o simulador NS-3 [30], variando aleatoriamente a posição, o número de UEs (entre 25 e 45) para

considerar diferentes condições de tráfego na célula e os valores do *CREO* usados pelo PeNB (entre 0 e 8 dBs) para modificar sua área de cobertura.

5.3.1.2.

Extração das características/atributos

Uma vez obtidas as amostras de treinamento, é necessário extrair os atributos de interesse. Este passo permitiu representar as observações através de valores numéricos. Para cada experiência (simulação realizada), os atributos extraídos foram: as condições de tráfego (em número de usuários), o valor de *CREO* utilizado pela PeNB e a perda de pacotes na camada *PDCP* (*Packet Data Convergence Protocol*) da tecnologia LTE para representar a qualidade do sistema [74].

5.3.1.3.

Aplicação dos algoritmos de aprendizagem (extração do modelo)

Neste passo, foi utilizada uma metodologia de aprendizagem supervisionada conhecido como regressão polinomial multivariável [75] para encontrar a relação entre as condições de tráfego na célula e o *CREO* (variáveis explicativas) com a perda de pacotes na camada *PDCP* de LTE (variável dependente).

Designando as variáveis, Z como a perda de pacotes, X o número de UEs ativos e Y como o valor de *CREO*, a perda de pacotes na camada *PDCP* é modelada pela equação 5.1:

$$Z = f(x, y), \quad (5.1)$$

A equação 5.2, apresenta um exemplo de uma função polinomial de segundo ordem,

$$f(x, y) = P_{00} + P_{10}x + P_{01}y + P_{20}x^2 + P_{11}xy + P_{02}y^2, \quad (5.2)$$

em que os coeficientes P_{10} , P_{01} são chamados de parâmetros de efeito linear, P_{20} , P_{02} de parâmetros de efeito quadrático e P_{11} de parâmetro de interação.

A partir das m experiências coletadas $\{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_m, y_m, z_m)\}$ são calculados os fatores P_{ij} que minimizam a soma dos resíduos dadas pela equação 5.3.

$$S = \min_p \sum_{i=1}^m [z_i - f(x_i, y_i)]^2 \quad (5.3)$$

Neste trabalho, o modulo extrator do modelo avalia vários graus (entre 1 e 4) para cada variável exploratória da função polinomial, selecionando aquela com os melhores resultados de aprendizado/ajuste.

5.3.1.4. Avaliação

Para avaliar a qualidade do modelo gerado, foi utilizado um conjunto de dados não utilizado durante a fase de treinamento. Para isso, foram computadas as métricas: soma dos quadrados devido aos resíduos (*sum square error*, *SSE*) e o coeficiente de determinação R^2 . O *SSE* determina o desvio total medido, que é dado pelas diferenças entre cada valor observado z_i e o valor previsto \hat{z}_i ,

$$SSE = \sum_{i=1}^m (z_i - \hat{z}_i)^2 \quad (5.4)$$

por outro lado, R^2 indica o quanto o modelo extraído consegue explicar os valores observados. Matematicamente R^2 é a correlação entre os valores de resposta e os valores de resposta previstos, sendo definida pela razão entre a soma dos quadrados da regressão (*sum of squares of the regression*, *SSR*) e o total da soma dos quadrados (*sum of squares*, *SST*). *SSR* é determinada pela equação 5.5.

$$SSR = \sum_{i=1}^m (\hat{z}_i - \bar{z})^2 \quad (5.5)$$

SST também é conhecido como a soma dos quadrados sobre sua média e é determinado pela equação 5.6:

$$SST = \sum_{i=1}^m (z_i - \bar{z})^2 \quad (5.6)$$

onde $SST = SSR + SSE$.

Finalmente, dadas as definições anteriores, R^2 é expressado como:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad (5.7)$$

R^2 toma valores entre 0 e 1, sendo que para valores perto do 1 indica que existe uma alta proporção da variação que pode ser explicada pelo modelo.

5.3.2. Seleção do CRE-offset

O valor do *CREO* utilizado pela PeNB é determinado de forma dinâmica seguindo o modelo extraído e procurando minimizar a perda de pacotes conforme as condições de tráfego na célula. Desta forma, o valor do *offset* é definido por:

$$CREO = \underset{y \in [y_1, y_2]}{\operatorname{argmin}} [f(x, y)] \quad (5.8)$$

em que x simboliza as condições de tráfego atual (número de UE) e y representa o *CREO* utilizado pela PeNB, pertencente ao intervalo de *offsets* ($y_1 = 0, y_2 = 8$) usados durante a fase de treinamento.

5.3.3. Implementação na arquitetura LTE

A integração da técnica de balanceamento de carga cognitivo foi realizada com base nos conceitos definidos para a arquitetura CCSON [31] descritos no capítulo 4. Nesta técnica, implementamos um grupo cognitivo composto por um agente cognitivo associado à PeNB e um agente cooperativo associado à MeNB que envia informações periódicas de desempenho e estado para a PeNB.

Na primeira fase, o agente associado a PeNB testa diversas configurações do valor do *CREO* para várias condições de tráfego na célula. Como resultado destes testes, são calculadas e armazenadas as estatísticas de desempenho a fim de construir o conjunto de dados que serão utilizadas para extrair o modelo de operação. Já na fase de otimização, a partir do modelo aprendido e das condições

de carga na célula, a PeNB calcula o valor do *CREO* que deve ser utilizado para otimizar o desempenho na rede de acesso.

A Figura 21 e a Figura 22 ilustram os componentes internos dos agentes associados aos elementos de acesso rádio.

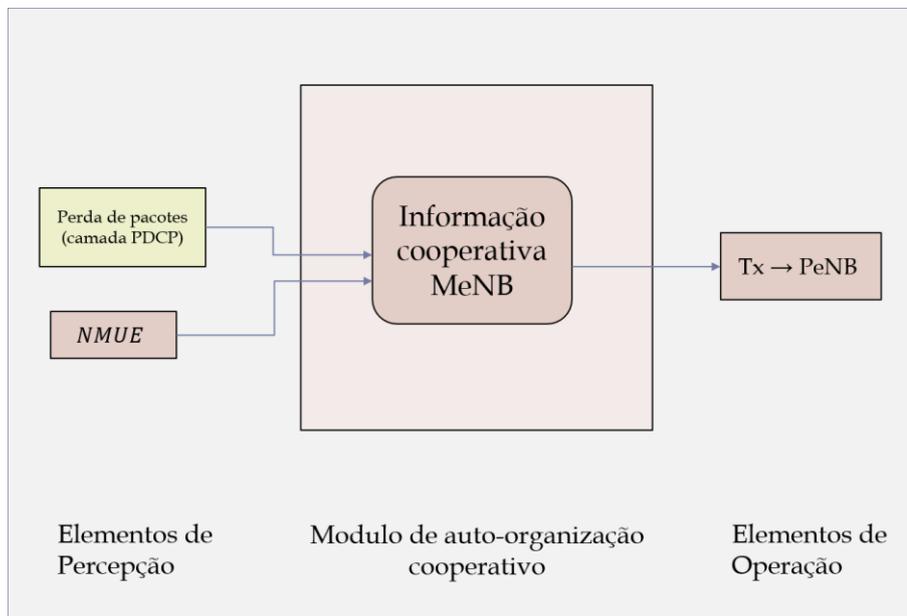


Figura 21. Agente cooperativo associado a MeNB

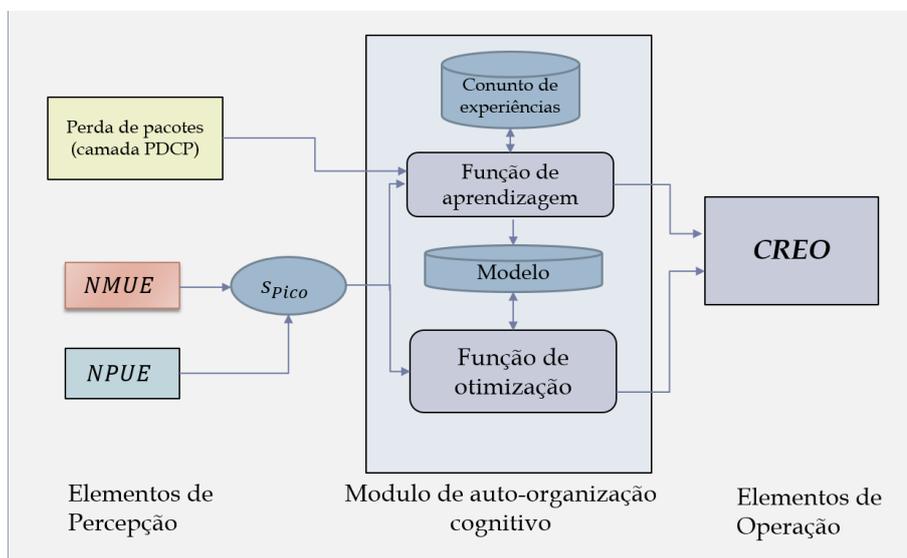


Figura 22. Agente cognitivo associado a PeNB

5.3.4. Avaliação de desempenho

Nesta seção apresenta-se a avaliação de desempenho da técnica proposta para o balanceamento de carga em redes heterogêneas LTE, especificamente para o cenário descrito na seção 5.2. Para isso, foram desenvolvidas simulações ao nível de sistema, comparando e analisando a perda de pacotes [74] na camada PDCP da tecnologia LTE para 3 casos:

- Um cenário homogêneo (sem a implementação das PeNB)
- Um cenário heterogêneo com seleção de célula baseado no máximo RSRP (MaxRSRP).
- Um cenário heterogêneo implementando a técnica proposta desenvolvida nesta seção, chamada de *Supervised Learning CRE Offset* (SL-CREO).

Para cada cenário, as condições de tráfego da célula foram variadas mudando o número de usuários que recebem uma transmissão de taxa constante UDP no enlace descendente.

5.3.4.1. Modelo extraído

O modelo extraído reflete a relação entre as condições de tráfego na célula e o valor de CREO utilizado pela PeNB, com a perda de pacotes na camada PDCP da tecnologia LTE (para o cenário estudado neste trabalho). Aqui, o modelo resultante foi caracterizado por uma função polinomial $f(x, y)$ de terceiro grau em x e quarto grau em y (equação 5.9), com um o coeficiente de determinação $R^2 = 0.715$.

$$\begin{aligned}
 & p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{30}x^3 + p_{21}x^2y \quad (5.9) \\
 & \quad + p_{12}xy^2 + p_{03}y^3 + p_{31}x^3y + p_{22}x^2y^2 + p_{13}xy^3 \\
 & \quad + p_{04}y^4
 \end{aligned}$$

$p_{00} = 5.711$	$p_{21} = 0.00605$
$p_{10} = -1.124$	$p_{12} = 0.0920$
$p_{01} = 12.597$	$p_{03} = 0.1097$
$p_{20} = 0.0448$	$p_{31} = 9.556e - 06$
$p_{11} = -0.615$	$p_{22} = -0.00105$

$p_{02} = -2.164$	$p_{13} = -0.000168$
$p_{30} = -0.000391$	$p_{04} = -0.005239$

Na Figura 23 e Figura 24 observa-se a gráfico em 3D e o gráfico de contorno do modelo extraído. Nessas Figuras, observa-se como o modelo permite prever os valores do *CREO* na PeNB que minimizariam a perda de pacotes média para o cenário de estudo. A razão para isso é que ao ampliar a cobertura PeNB seria possível descarregar uma maior quantidade de usuários da MeNB para a PeNB diminuindo a congestão e reduzindo ao mesmo tempo a perda de pacotes na célula. No entanto, também é possível observar as limitações para os valores do *CREO*, dado que se a cobertura da PeNB for incrementada além de determinado ponto, a interferência produzida pela MeNB afetará fortemente aos usuários na sua borda.

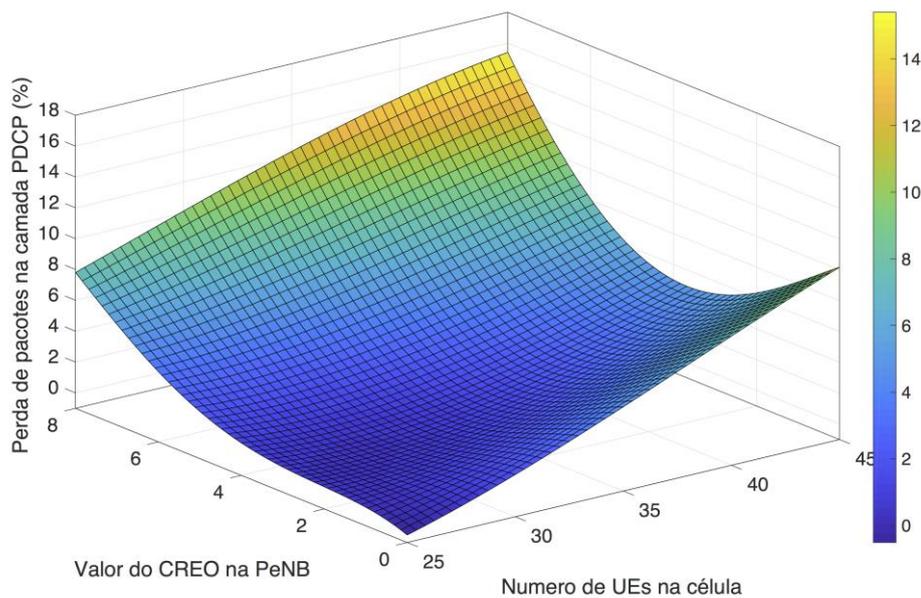


Figura 23. Modelo extraído: gráfico em 3D

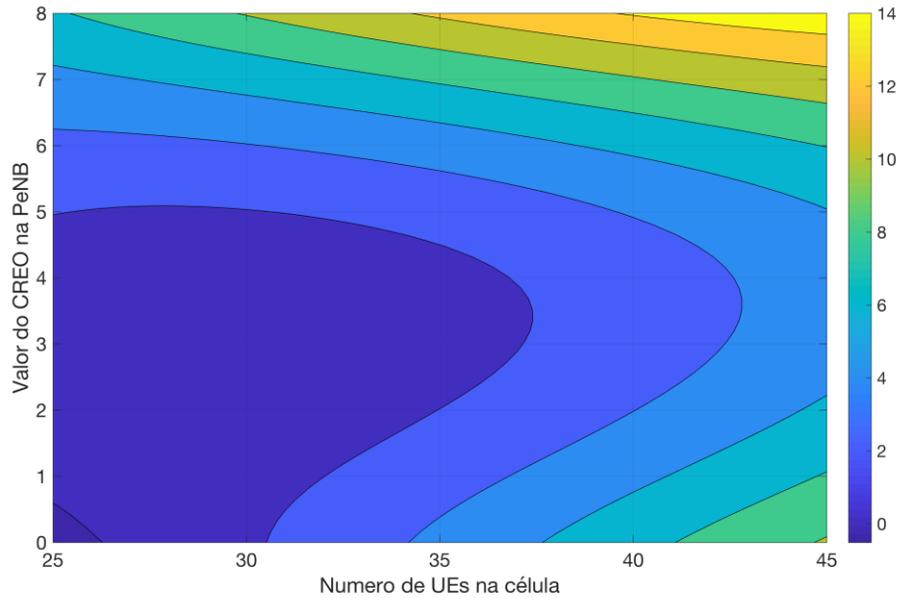


Figura 24. Modelo extraído: gráfico de contorno

5.3.4.2. Perda de pacotes na camada PDCP

A Figura 25 mostra a relação entre a perda de pacotes PDCP média e o número de UEs ativos na célula (macro + pico eNB) para os três casos propostos. Observa-se como a perda de pacotes é incrementada com o incremento da carga da célula. A Figura 25 mostra também como a implementação do ambiente heterogêneo beneficia o desempenho médio da célula e como através da introdução do mecanismo proposto a perda de pacotes diminui ainda mais. Neste caso, para uma quantidade de 40 e 45 UEs, através da técnica SL-CREO experimenta-se uma diminuição na perda de pacotes de 7.22% para o 4.67% e do 10.89% para 7.5%, respectivamente, quando comparado com o uso da técnica tradicional MaxRSRP.

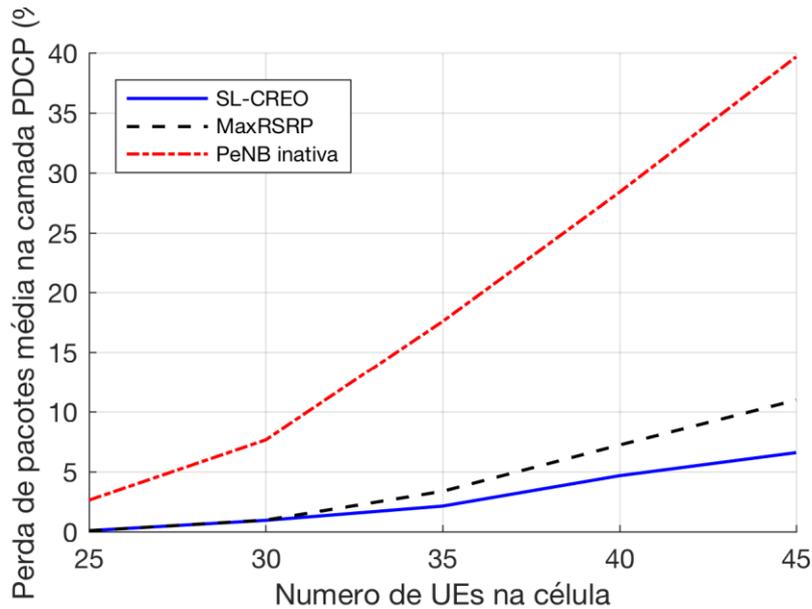


Figura 25. Perda de pacotes na camada PDCP média Vs número de UEs ativos na célula

Na Figura 26 ilustra-se a função de distribuição acumulada empírica para a perda de pacotes quando a célula se encontra em uma situação de congestionamento (40 UEs). Neste caso, os três cenários apresentam perdas devido ao fato de que não existem recursos disponíveis para atender a demanda de tráfego. No entanto, observa-se como o esquema proposto consegue melhorar o desempenho médio da célula.

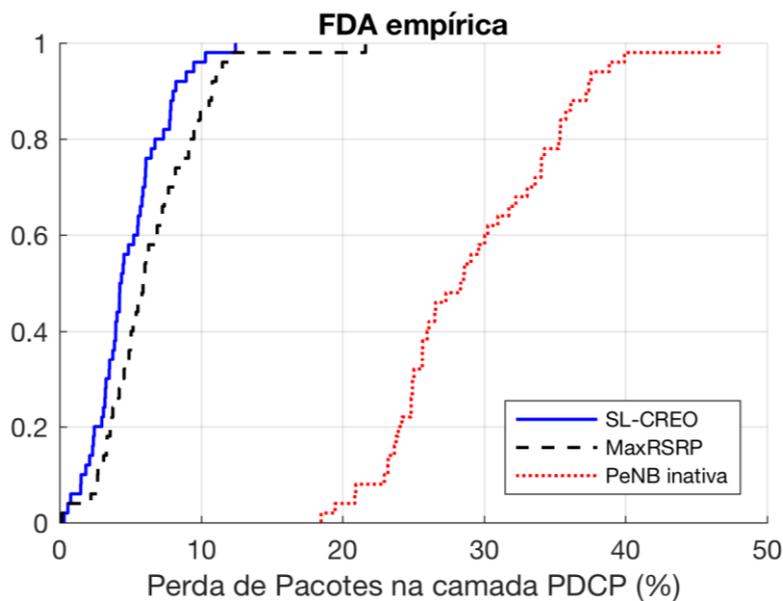


Figura 26. FDA empírica para a perda de pacotes média na camada PDCP (40 UEs)

5.3.4.3. Utilização de recursos na PeNB

A Figura 27 compara a utilização dos blocos de recursos físicos (*Physical Resource Block, PRB*) com e sem a implementação da técnica SL-CREO para diferentes condições de tráfego. Observa-se como a implementação do CREO dinâmico permite incrementar a utilização de recursos na PeNB (sem sobrecarregá-la) dado que uma maior quantidade de UEs são transferidos da MeNB para a PeNB. Neste caso os recursos liberados na MeNB são utilizados por outros UEs, diminuindo a perda de pacotes e melhorando consequentemente o desempenho da célula.

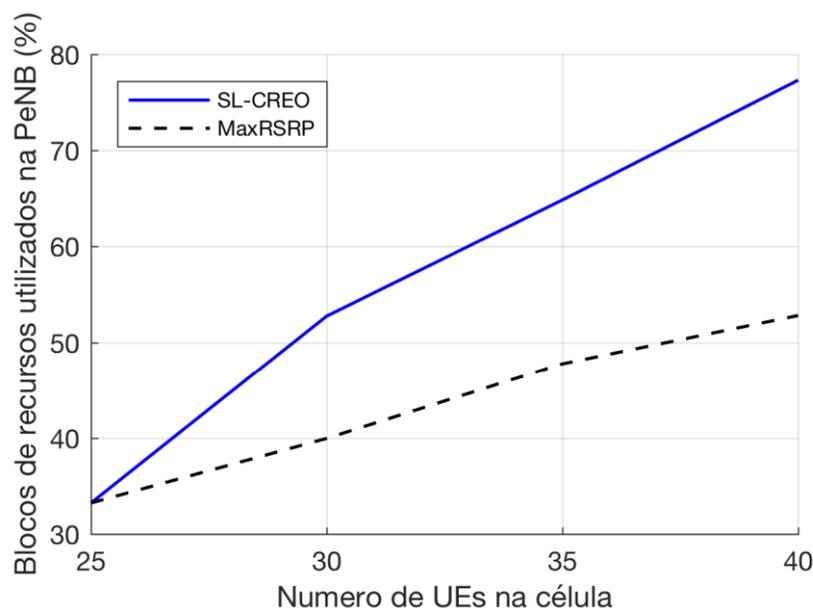


Figura 27. Utilização de blocos de recursos na PeNB

5.4.Técnica 2: balanceamento de carga cognitivo através de uma abordagem por aprendizagem por reforço

Como descrito na seção 3.2, a aprendizagem por reforço (*Reinforcement Learning, RL*) é um tipo de problema encontrado por um agente autônomo que deve aprender como agir em um ambiente dinâmico através de iterações de tentativa e erro [61]. Nesta abordagem, cada pico-eNB (PeNB) é modelada como um agente que percebe o estado do entorno e interage com ele ao modificar seu próprio CRE *offset* (CREO) em busca de maximizar o desempenho da célula [32], tal como se ilustra na Figura 28. Assim, o processo de decisão foi modelado como

um processo de Markov (MDP) com um vetor de estados S (descrito na seção 5.4.2) a cada instante t . O agente escolhe uma ação $a \in A = \{a_1, a_2, \dots, a_{|A|}\}$ e como resultado de dita ação o ambiente retorna uma recompensa R . A partir desta informação, o agente atualiza localmente a cada iteração os valores da função de estado-ação $Q(s, a)$ através de um processo de diferenças temporais (livre-de-modelo). Como resultado desta exploração do ambiente, o agente aprende ao longo do tempo qual ação resulta na melhor recompensa para cada possível estado percebido.

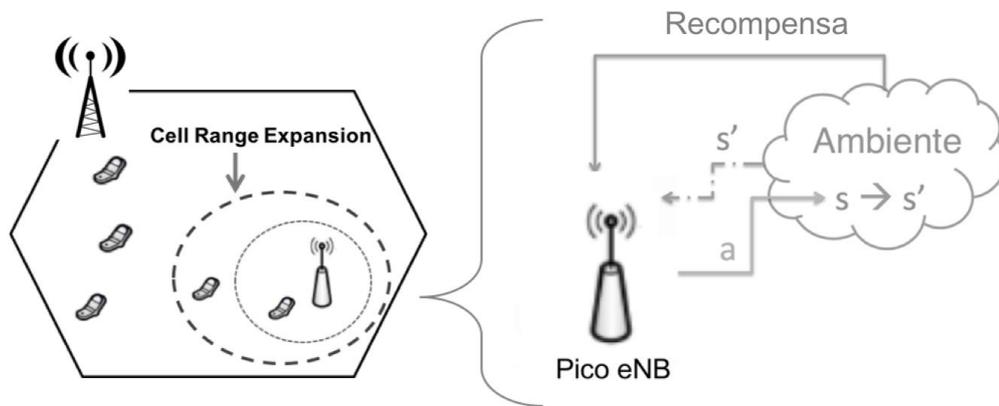


Figura 28. Pico eNB como agente de aprendizagem

Dado que o cenário de estudo apresenta um conjunto de variáveis contínuas, no desenvolvimento desta técnica de balanceamento de carga, utilizamos uma adaptação do algoritmo Q -Learning conhecido como *Fuzzy Q-Learning* (FQL) [76]. O FQL permite flexibilidade na discretização e segmentação das variáveis utilizadas para a representação dos estados e das ações que devem ser aplicadas às regras definidas pelo sistema de inferência *Fuzzy* (*Fuzzy Inference System*, FIS).

A seguir apresenta-se o algoritmo FQL usado para realizar o processo de aprendizado do *CREO* utilizado pela PeNB, além dos detalhes de implementação e análise de desempenho da técnica de balanceamento de carga cognitiva desenvolvida.

5.4.1. Algoritmo Fuzzy Q-Learning

Tal como mencionado na seção 3.2.5, o algoritmo Q -Learning baseia-se na estimação dos valores estado-ação da função $Q(s, a)$, que representam a

recompensa esperada que pode ser recebida quando realizada a ação a no estado s . O algoritmo *Fuzzy Q-Learning* (FQL) trata diretamente com conjuntos de valores contínuos, sendo possível realizar a divisão das variáveis contínuas em variáveis de estado s com valores linguísticos *Fuzzy* e representando a tabela de valores $Q(s, a)$ através de um conjunto de regras r_i ($i = 1, 2, \dots, N$). Desta forma, dado um vetor de entrada de estado X e o conjunto de ações possíveis $A = \{a_1, a_2, \dots, a_{|A|}\}$, a representação das regras FQL estará dada por:

Se s é r_i , então ação= a_1 com $q(r_i, a_1)$

Ou ação= a_2 com $q(r_i, a_2)$

...

Ou ação= $a_{|A|}$ com $q(r_i, a_{|A|})$

na qual s representa o estado percebido e r_i uma das possíveis regras ativadas. Neste caso, observa-se como cada regra do FIS apresenta múltiplos possíveis consequentes associados com um valor de qualidade $q(r_i, a_j)$. Esta representação, permite caracterizar a função Q mediante uma matriz $N * |A|$, onde N representa o número de regras e $|A|$ o número de possíveis ações.

Neste caso, o objetivo do algoritmo FQL é obter o melhor consequente (através de uma estratégia de exploração) para cada regra no FIS conforme um conjunto de ações/consequentes possíveis.

De forma geral, o algoritmo FQL explora a relação de pares antecedentes/consequentes (estado/ação) do FIS para determinar as políticas que maximizam suas recompensas. Para isso, em cada iteração de exploração/aprendizagem é realizado um ciclo resumido nos seguintes passos:

- 1) Perceber o estado $s(t)$ do ambiente e realizar o processo de fuzzificação.
- 2) Para cada regra ativada no FIS (por causa do estado percebido), escolher um consequente conforme a política de exploração/otimização.
- 3) Calcular a ação resultante pelo FIS e aplicá-la no ambiente, deixando o sistema evoluir para $t = t + 1$
- 4) Observar o novo estado $s(t + 1)$ do ambiente e a recompensa $R(t + 1)$ recebida.

- 5) Aprender da experiência realizada (transição), atualizando a função $Q(s,a)$

O algoritmo 1 apresenta o passo a passo do processo de aprendizagem realizado pelo FQL.

Algoritmo 1: Algoritmo *Fuzzy Q-Learning*

Entradas: $\epsilon, \gamma, \lambda$

Saída: tabela de valores-q (estado, ação).

Início:

Para cada passo de tempo.

- 1: Perceber o estado $s(t)$ do ambiente e realizar o processo de Fuzzificação.
- 2: Para cada regra ativada no *FIS* (por causa do estado percebido), escolher um conseqüente conforme a política de exploração/otimização.

$$a_i = \begin{cases} \underset{j}{\operatorname{argmax}}[q(r_i, a_j)], & \text{com probabilidade } (1 - \epsilon) \\ a_j \text{ aleatoria,} & \text{com probabilidade } (\epsilon) \end{cases}$$

- 3: Calcular a ação global a inferida pelo controlador *Fuzzy* e calcular seu valor de qualidade global $Q(s, a)$:

$$Q(s, a) = \frac{\sum_{i=1}^N \alpha_i(s) q[r_i, a_i]}{\sum_{i=1}^N \alpha_i(s)},$$

em que $\alpha_i(s)$ é o grau de ativação da regra r_i e N o número de regras.

- 4: Aplicar a ação a indicada pelo *FIS*, deixar o sistema evoluir para $s(t + 1)$
- 5: Observar o valor de recompensa $R(t + 1)$ e calcular o valor do novo estado:

$$V(s(t + 1)) = \frac{\sum_{i=1}^N \alpha_i(s(t + 1)) \max_j [q(r_i, a_j)]}{\sum_{i=1}^N \alpha_i(s(t + 1))}$$

- 6: Calcular a diferença entre $Q(s(t), a)$ e $Q(s(t + 1), a)$:

$$\Delta Q = [R(t + 1) + \gamma V(s(t + 1))] - Q(s(t), a)$$

onde γ representa o fator de desconto.

7: Atualizar a tabela de valores-q através de diferenças temporais:

$$q(r_i, a_i) = q(r_i, a_i) + \lambda \Delta Q \frac{\alpha_i(s)}{\sum_{i=1}^N \alpha_i(s)}$$

onde λ é o fator de aprendizagem.

Fim

Na sequência são descritos os componentes do MDP que modelam o processo de aprendizagem por reforço: espaço de estados, ações e função de recompensa.

5.4.2.

Definição de estados, ações e recompensas

5.4.2.1.

Estados

A técnica aqui desenvolvida, visa modificar o *CREO* da PeNB conforme as condições de tráfego na célula. Neste caso, a tarefa de aprendizagem é realizada na PeNB utilizando informações relevantes do estado da célula, sendo estas definidas pelo vetor:

$$s = [NMUE, NPUE, SINR_5], \quad (5.10)$$

no qual *NMUE* e *NPUE* representam a carga da macro e da pico-eNB em termos do número de usuários, *SINR₅* faz referência ao Percentil 5 (*5th Percentile*) do *SINR* dos usuários localizados na pico célula, valor comumente utilizado para determinar o desempenho dos usuários da borda da mesma.

As Figuras 29 e 30 ilustram a definição dos conjuntos *Fuzzy* de cada uma das variáveis utilizadas. A Figura 29, apresenta as funções de pertinência para as variáveis *NMUE* e *NPUE*, já a Figura 30 apresenta as funções de pertinência utilizadas para descrever a variável *SINR₅*.

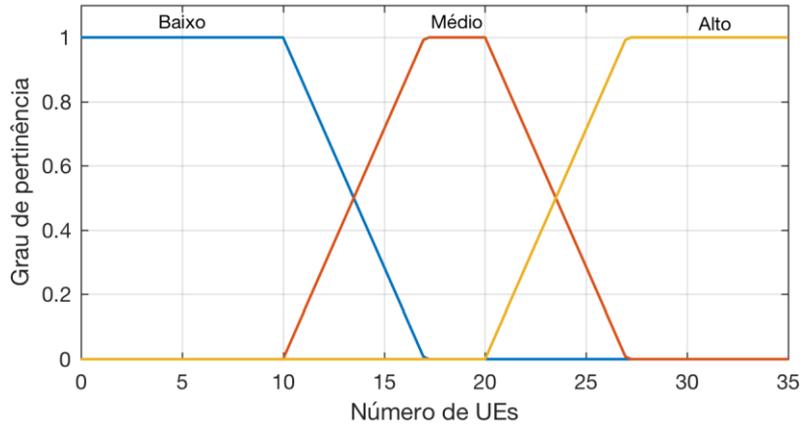


Figura 29. Funções de pertinência para a variável de estado referente ao número de usuários na MeNB/PeNB.

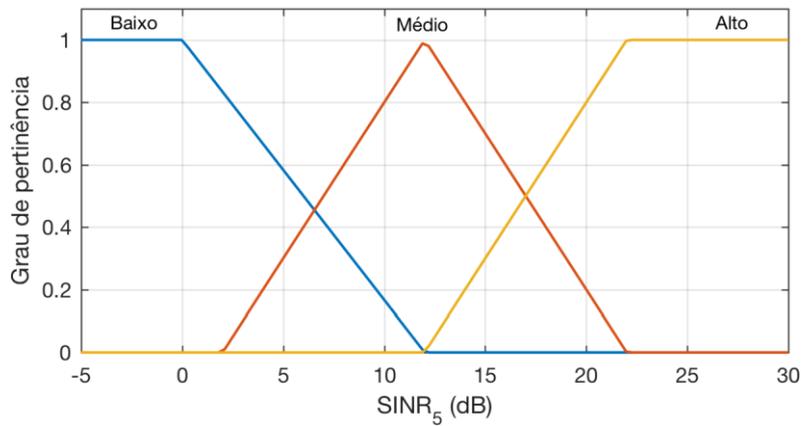


Figura 30. Funções de pertinência para a variável de estado que representa a qualidade dos UEs na borda da pico célula

A partir da anterior segmentação das variáveis de estado, o sistema de inferência define-se por um número de 27 regras possíveis (resultado da combinação dos antecedentes). Estas regras serão exploradas conforme se apresentem durante a fase de treinamento do sistema (Seção 5.4.4).

5.4.2.2. Ações/consequentes.

O conjunto de ações possíveis consiste nos possíveis conjuntos *fuzzy* selecionáveis em cada regra do *FIS*. A Figura 31 ilustra a definição destes conjuntos bem como as funções de pertinência utilizadas.

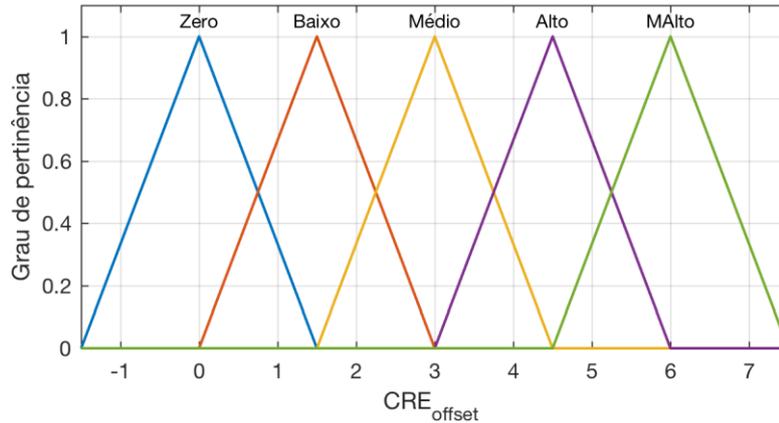


Figura 31. Funções de pertinência para a variável de saída $CREO$

5.4.2.3. Função de recompensa

A recompensa no sistema FQL proposto é dada pela vazão média por usuário na célula. Definindo U como o número total de usuários com uma comunicação ativa na célula em um instante t , o valor de vazão é descrito pela equação 5.11, a seguir:

$$V_t = \sum_{u=1}^U \frac{v_t(u)}{U} \quad (5.11)$$

em que $v_t(u)$ representa a vazão do u -ésimo usuário ($u = 1, 2, \dots, U$).

5.4.3. Implementação na arquitetura LTE

A integração da técnica de balanceamento de carga cognitivo foi realizada com base nos conceitos definidos para a arquitetura CCSON [31] descritos no capítulo 3. Especificamente, foi implementado um grupo cognitivo composto por um agente cooperativo associado à MeNB e um agente cognitivo associado à PeNB. Estes agentes compartilham entre eles informações de estado e desempenho através da interface X2 conforme seja requerido pela técnica de balanceamento de carga e o estado da célula. A Figura 32 e a Figura 33 ilustram os componentes funcionais dos agentes envolvidos neste processo.

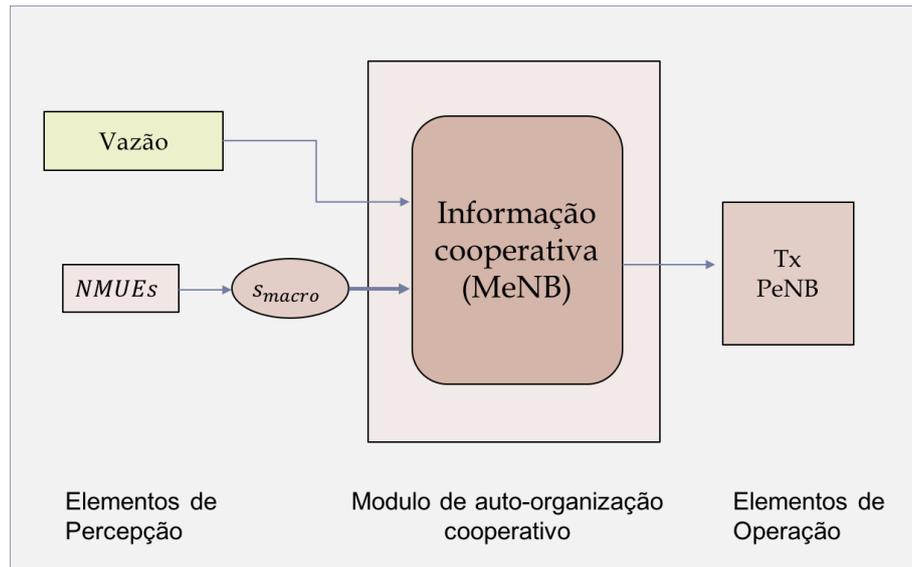


Figura 32. Agente cooperativo associado a MeNB

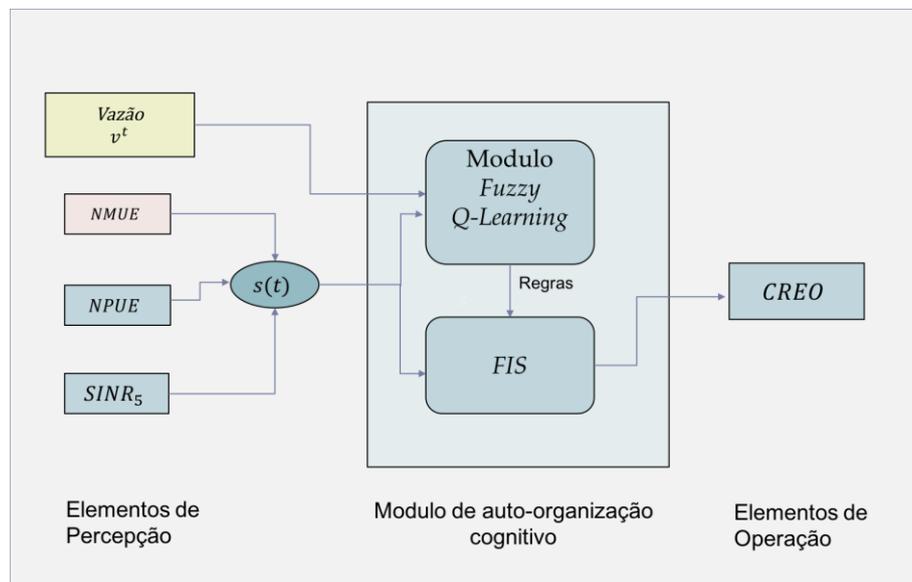


Figura 33. Agente cognitivo associado a PeNB

5.4.4. Simulação do processo de aprendizagem

O cenário descrito na seção 5.2 foi simulado através do simulador de código aberto NS-3, o qual foi integrado com a ferramenta Matlab onde foi implementado o algoritmo de aprendizado automático. Este algoritmo opera no agente associado à PeNB, modificando dinamicamente o *offset* (*CREO*) que determina a cobertura da pico-célula.

Durante a etapa de aprendizagem foram considerados 250 episódios de treinamento com o objetivo de simular diversas intensidades de tráfego (estados)

na rede. Para isso, em cada um dos episódios de treinamento, variamos aleatoriamente o número de UEs no intervalo [30,45] e seus posicionamentos na célula. Nos experimentos de simulação, cada UE recebe uma transmissão UDP no enlace descendente com taxa constante de 512 kbps.

Inicialmente, todos os valores da tabela Q foram inicializadas em zero. Isso para representar que a PeNB não possui informação *a priori* sobre as políticas de ações do problema.

Cada episódio de treinamento abrange 3 iterações de aprendizagem nas quais o $CREO$ é modificado conforme uma política de exploração ϵ -greedy (passo 2 no algoritmo 1). Durante esta primeira fase de treinamento utilizamos uma taxa de exploração alta ($\epsilon = 0.7$) para que o sistema explore amplamente os consequentes das regras do sistema *fuzzy*, atualizando constantemente os valores da tabela Q . Os valores da taxa de aprendizagem λ e fator de desconto γ foram 0.2 e 0.8, respectivamente.

Na seguinte fase, a taxa de exploração é diminuída ($\epsilon = 0$) para que o sistema aproveite o conhecimento adquirido (otimização). Na sequência, apresenta-se a validação da técnica aqui proposta.

5.4.5. Avaliação de desempenho

Nesta seção, apresenta-se a avaliação de desempenho da técnica proposta para o balanceamento de carga em redes heterogêneas LTE. Para isso empregamos as métricas de desempenho de QoS: perda de pacotes na camada PDCP e a vazão média por usuário na célula. Os resultados obtidos pela técnica proposta nesta seção ($FQL-CREO$) são comparados com aqueles obtidos para o esquema de seleção de célula baseado no máximo RSRP (MaxRSRP), com a técnica SL-CREO (apresentada na seção 5.3) e com a técnica ABC (*Adaptive Bias Configuration*) desenvolvida por Tiang e colaboradores [77].

A técnica ABC tem como objetivo modificar dinamicamente o $CREO$ da PeNB através de um algoritmo analítico e iterativo a partir de realimentações de desempenho do sistema. Tal ajuste é realizado paulatinamente ao valor atual do $CREO$ através da adição de um termo $\Delta CREO$.

5.4.5.1. Resultados da aprendizagem

Na Figura 34 observa-se o gráfico da superfície que representa a relação aprendida pela PeNB entre a carga da MeNB (em número de UEs ativos) e a sinal-ruído-interferência ($SINR$) experimentada pelos usuários da borda da PeNB em relação com o valor do $CREO$ que produz o melhor desempenho na célula. Neste caso de aprendizado, observa-se como o PeNB identificou a possibilidade de aumentar o valor do $CREO$ quando a carga do tráfego na MeNB alcança valores altos sempre que a $SINR$ de seus usuários na borda apresente valores acima de 20 dB. Por outro lado, quando a $SINR$ dos usuários da borda da PeNB experimente valores abaixo do 0 dBs, o $CREO$ deve ser mantido no menor valor para não prejudicar o desempenho da célula. De forma similar, a Figura 35 ilustra o gráfico de superfície (aprendido pela PeNB) para as variáveis $NPUE$, $SINR_5$ e $CREO$ que resultam no melhor desempenho da célula.

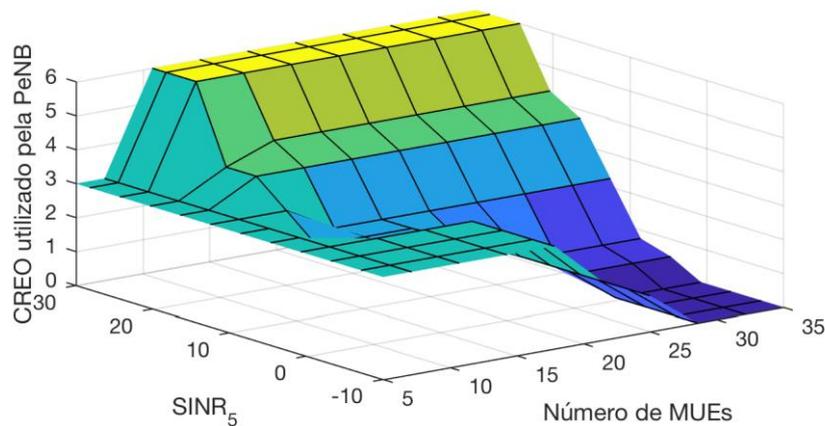


Figura 34. Relação aprendida entre as variáveis $NPUE$, $SINR_5$ e $CREO$ que produzem o melhor desempenho na célula

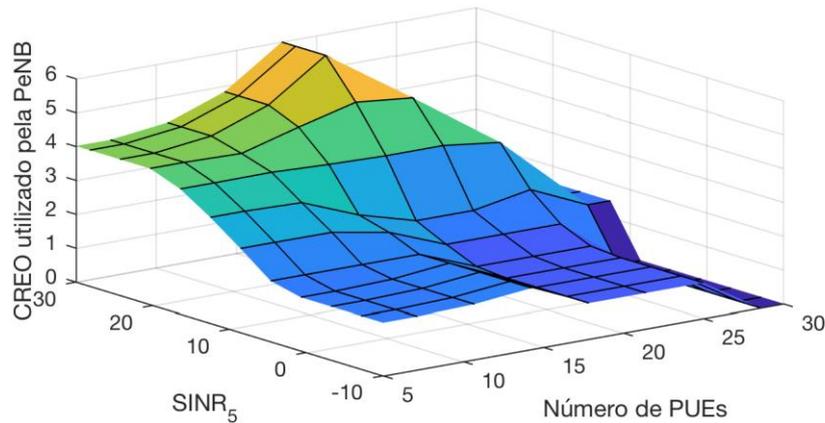


Figura 35. Relação aprendida entre as variáveis NPUE, $SINR_5$ e $CREO$ que produzem o melhor desempenho na célula

5.4.5.2. Avaliação das métricas de QoS

A Figura 36 mostra a relação entre a perda de pacotes PDCP média e o número de UEs ativos na célula (macro + pico). A Figura 37 ilustra a função de distribuição acumulada empírica de probabilidade (FDA) especificamente para o cenário com 40 UEs ativos na célula. Em ambas figuras se observa a efetividade da técnica utilizada para aprender os valores do $CREO$ que melhoram o desempenho da célula conforme as condições de tráfego da mesma.

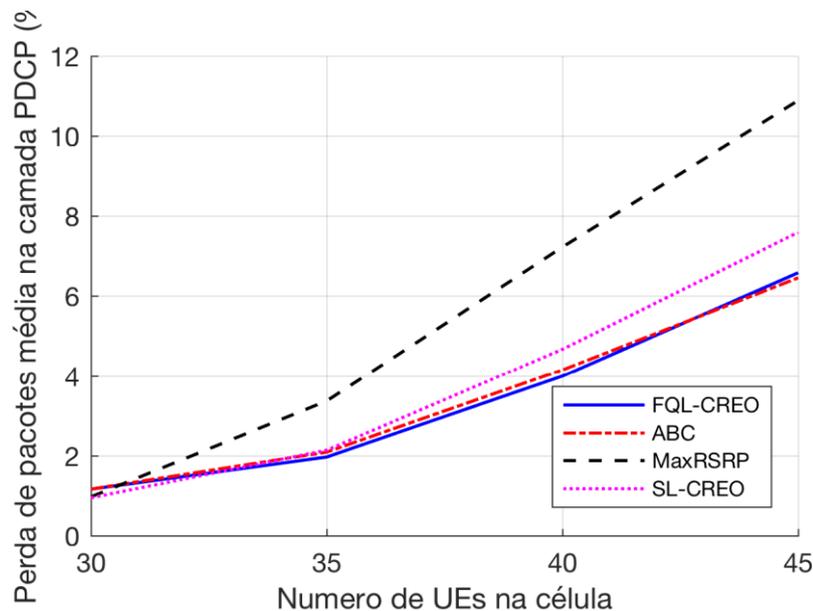


Figura 36. Perda de pacotes média na camada PDCP vs. Número de UEs ativos na célula

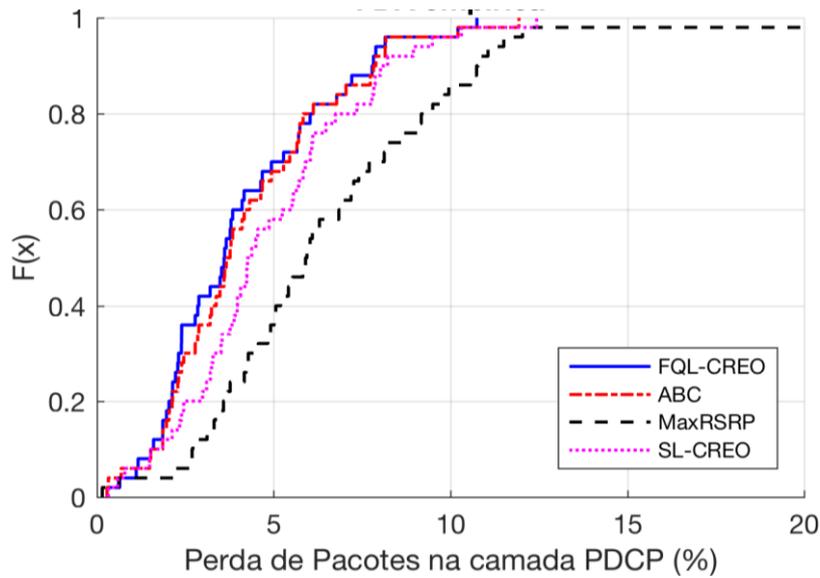


Figura 37. FDA empírica para a perda de pacotes na camada PDCP (40 UEs)

Nas Figura 38 e Figura 39 observam-se os efeitos da perda de pacotes sobre a vazão média por usuário em relação ao número de UEs ativos na célula e a FDA empírica para o cenário de simulação com 40 UEs.

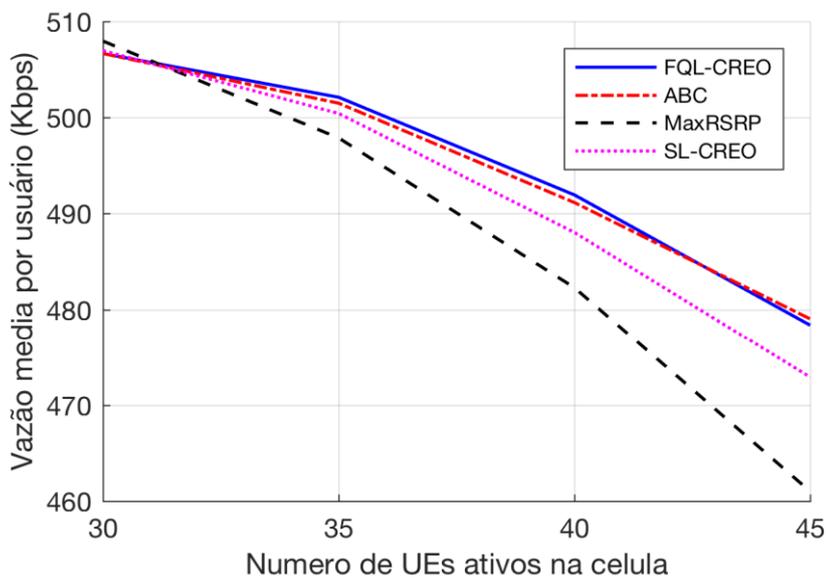


Figura 38. Vazão média por UE na camada PDCP vs. Número de UEs ativos na célula

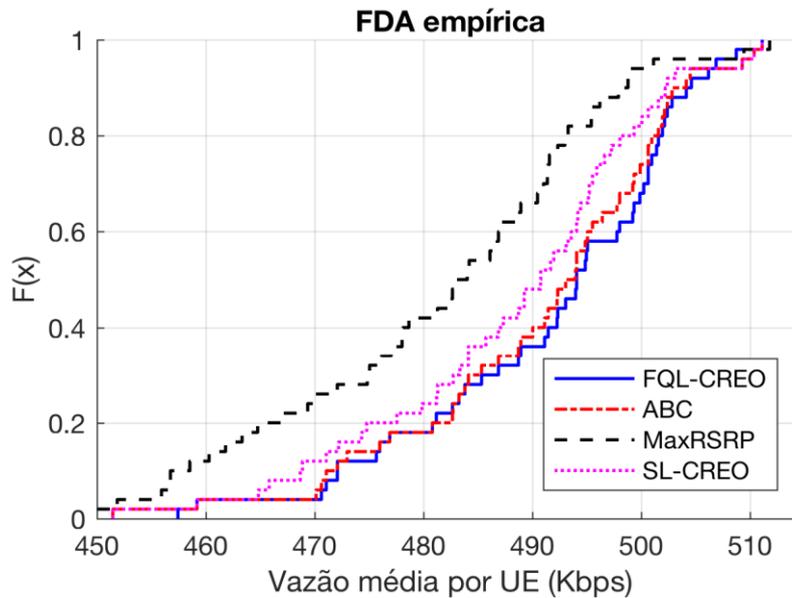


Figura 39. FDA empírica para a vazão média por UE na camada PDCP (40 UEs)

Da Figura 38 e Figura 39, consegue-se identificar os benefícios da técnica *FQL-CREO*. Esta seleção dinâmica do *offset* permite um melhor desempenho já que uma maior quantidade de UEs consegue ser transferida para a PeNB de forma controlada.

Observou-se nos resultados das simulações que a técnica FL-CREO teve um melhor desempenho quando comparada com a abordagem SL-CREO. Isto dado que a técnica FL-CREO apresenta uma análise mais detalhada do estado atual da célula, levando em conta separadamente o estado do tráfego tanto na MeNB quanto na PeNB para a toma da decisão do valor do CREO.

Nos experimentos de simulação, a técnica FQL-CREO igualou o desempenho conseguido pela técnica ABC usada como referência, mostrando a eficácia dos mecanismos de aprendizagem para sustentar funcionalidades SON. Neste caso, destacamos que o resultado tanto do SL-CREO quanto do FL-CREO foram obtidos em um passo de simulação, enquanto que a técnica ABC precisou de vários ajustes consecutivos no valor do **CREO** (incrementos ou decrementos) até chegar no valor que produz os melhores resultados.

5.5. Observações finais

Neste capítulo foram apresentadas duas técnicas de balanceamento de carga para redes móveis heterogêneas, através de abordagens cognitivas e aprendizado automático. Os resultados das simulações mostraram a eficiência das técnicas desenvolvidas para aprender a modificar os valores do *CREO* segundo as condições de tráfego na célula. Desta forma, foram obtidas melhores condições de balanceamento da carga, reduzindo o congestionamento nas MeNB e, conseqüentemente, melhorando as métricas de QoS nos usuários finais.

Na Tabela 2 apresentam-se as principais características das técnicas propostas neste capítulo.

Tabela 2. Resumo das técnicas cognitivas desenvolvidas para o balanceamento de carga em redes móveis heterogêneas

Características	Técnicas cognitivas para o balanceamento de carga	
	Técnica 1 (Seção 5.3)	Técnica 2 (Seção 5.4)
Decisão	Modificação do valor de <i>CREO</i> utilizado pela PeNB para incrementar sua cobertura	
Elemento que toma a decisão	Agente associado à PeNB	
Formulação do problema	Aprendizagem Supervisionado	Aprendizagem por reforço
Método	Regressão linear multivariada	Diferenças temporais
Algoritmo	<i>Least Squares fitting</i>	<i>Fuzzy Q-Learning</i>
Tipo de solução	Baseada em modelo (modelo polinomial)	Livre de modelo (tabela de regras)

Nas duas técnicas apresentaram-se resultados positivos no desempenho da célula. No entanto, observamos vantagens importantes no uso da **técnica 2**, sustentada na abordagem de aprendizagem por reforço. Isto principalmente devido à definição do problema, que assume que os dispositivos de acesso não contam com informações iniciais sobre o meio no qual vão operar. Nesse sentido, a **técnica 1**, fundamentada em um tipo de aprendizagem supervisionado, precisa coletar informações suficientes e relevantes da região de interesse antes de iniciar processos de otimização. Isto significa que precisam ser testadas múltiplas configurações nos parâmetros de rede (*CREO*) em múltiplas situações de tráfego

na célula para conseguir extrair um modelo que permita determinar as ações que otimizam o desempenho da célula.

Por outro lado, a abordagem baseada na aprendizagem por reforço (técnica 2) não precisa de um conjunto de dados de treinamento para operar. Neste caso, a aprendizagem é realizada a partir de iterações do sistema com o ambiente, no qual o sistema aprende através da experiência e as recompensas recebidas. Isto permite que o sistema aprenda as políticas (valores do *CREO*) dos estados (situações de tráfego) na medida em que eles se apresentam, sem precisar explorar todos os possíveis estados. Na técnica 2, o uso do algoritmo *Fuzzy Q-Learning* permitiu segmentar as variáveis de estado contínuas representando o mecanismo de decisão a partir de regras *fuzzy*. Dentre as limitações observadas, menciona-se que esta técnica é dependente da predefinição do número de conjuntos *Fuzzy* e as funções de pertinência usadas nos antecedentes e consequentes que compõem as regras. Além disso, a técnica apresenta limitações práticas enquanto ao número de variáveis de entrada usadas pelo FIS. Isto é devido a que o aumento de variáveis aumentaria exponencialmente o número de regras a ser aprendidas/exploradas pelo sistema. Contudo, consideramos que a abordagem por reforço representa a melhor aproximação para o desenvolvimento da técnica SON de balanceamento de carga estudada nesta tese.

A partir das observações acima, no capítulo seguinte continuaremos explorando o problema de balanceamento de carga cognitivo em redes de acesso móveis heterogêneas, para uma rede de acesso que considera tráfego multi-serviço e medidas de qualidade centradas no usuário.

6

Técnica cognitiva de balanceamento de carga em redes móveis heterogêneas multi-serviço

As redes móveis atuais vêm enfrentando novos desafios causados pela massificação de serviços multimídia tais como: voz sobre IP, jogos interativos, vídeo em streaming, compartilhamento de recursos e descarga de arquivos, entre outros. Essa situação tem levado aos operadores de rede a prestar cada vez mais atenção no desempenho do sistema, não apenas do ponto de vista do sinal e da cobertura, mas também do ponto de vista da qualidade que os usuários experimentam nos serviços recebidos [78] [79] [80].

As técnicas de balanceamento de carga cognitivas desenvolvidas no capítulo anterior [33][32] mostraram sua efetividade no processo de aprendizado para expandir dinamicamente a cobertura das pico-células conforme as condições de tráfego, balanceando a carga e melhorando o desempenho da célula. Porém, estas técnicas foram avaliadas apenas para situações de tráfego com taxa constante e com o objetivo de otimizar o desempenho da rede em termos da vazão da célula.

Neste capítulo, apresenta-se o desenvolvimento de uma nova técnica de balanceamento de carga cognitiva em redes heterogêneas. Esta técnica leva em consideração a influência de diferentes classes de serviço sobre o desempenho da rede e a qualidade de experiência (QoE, *quality of experience*) percebida pelos usuários finais.

A seguir, na Seção 6.1, apresentam-se as modificações realizadas na técnica *cell range expansion (CRE)* para incluir capacidades de otimização multi-serviço. Na sequência, a Seção 6.2 detalha as métricas de qualidade utilizadas para avaliar o desempenho desde a perspectiva dos usuários conforme o tipo de tráfego utilizado. Na seção 6.3, descreve-se o desenvolvimento do mecanismo de aprendizagem autônomo para o balanceamento de carga em redes móveis multi-serviço, logo na seção 6.4, detalha-se sua implementação nas arquiteturas da tecnologia LTE. Na seção 6.5, descreve-se o cenário simulação e a processo

realizado na etapa de aprendizagem e, finalmente, na seção 6.6, apresentam-se os resultados obtidos através de processos de simulação.

No decorrer deste capítulo serão apresentadas diferentes variáveis, as quais estão sintetizadas na Tabela 3 a fim de esclarecer sua leitura no texto.

Tabela 3. Glossário das principais variáveis utilizadas neste capítulo

Variável	Definição	Seção
K	Conjunto de classes de serviços	Seç. 6.1
$CREO$	Vetor de <i>offsets</i> de expansão de cobertura	
$CREO_k$	Valor de <i>offset</i> utilizado pelos usuários com serviço k	
E	Conjunto de estações rádio-base selecionáveis por um terminal de usuário.	
U	Conjunto de terminais de usuário na célula.	Seç. 6.2
MOS_k	<i>Mean opinion score</i> para o serviço da classe k	
S_{macro}	Variáveis de estado percebidas pela MeNB	Seç. 6.3.2
$NRAB_{macro}^k$	Número de comunicações (<i>radio access bearers</i>) ativas com tráfego pertencente a classe de serviços k	
ρ_{macro}	Porcentagem de utilização de blocos de recursos (recursos rádio) na MeNB	
S_{pico}	Variáveis de estado percebidas pela PeNB	
CQI_{5-pico}^k	Qualidade do estado do canal dos usuários da borda da pico-célula com tráfego pertencente a classe de serviços k .	
ρ_{pico}	Porcentagem de utilização de blocos de recursos na PeNB	
$s_k(t)$	Subestado analisado na PeNB pela subtarefa k	Seç. 6.3.4
$\pi_k(s_k)$	Política de ação da subtarefa k	
$\Delta MOS_k(t)$	Diferencia da medida da MOS do serviço k em dos tempos de observação consecutivos.	
R_k	Função de recompensa da subtarefa k	
w_k	Vetor de pesos que representa a importância relativa de cada tipo de serviço para a subtarefa k	
FIS_k	Sistema de inferência <i>Fuzzy</i> na subtarefa k .	Seç. 6.3.5
N_k	Número de regras no FIS_k	
$r_{n,k}$	Regra n no FIS_k	
O	Conjunto de possíveis consequentes para cada regra	
$q_k[n_k, o_j]$	Valor de qualidade de cada possível consequente o_j para a regra n no FIS_k	
$Q_k(s_k, CREO_k)$	Valor de qualidade global	
ϵ	Taxa de exploração	

6.1. Expansão virtual de cobertura multi-serviço

Como apresentado no capítulo 5, a técnica *CRE* permite incrementar a cobertura das pico-células sem incrementar suas potências de transmissão. Para isso, cada PeNB comunica um valor de *offset* aos terminais que é somado ao valor da potência recebida (de cada PeNB) utilizado pelos algoritmos de seleção de célula nos terminais de usuário. Como resultado desta operação, incrementa-se a possibilidade de que alguns terminais se associem com as pico-células no lugar das macro-células.

Dada a necessidade de desenvolver mecanismos que levem em consideração as características multi-serviço das redes móveis atuais, propomos uma modificação na técnica *CRE*, visando otimizar a utilização dos recursos rádio enquanto se cumprem com os requisitos de qualidade esperados para os diferentes tipos de serviços oferecidos.

A ideia desenvolvida consiste em agrupar os usuários em um conjunto de $|K|$ classes de serviços $k \in K = \{k_1, k_2, \dots, k_{|K|}\}$, designando para cada um desses grupos um *offset* diferenciado através do vetor $\overline{CREO} = [CREO_1, CREO_2, \dots, CREO_{|K|}]$. Desta forma, procura-se ter uma maior granularidade e flexibilidade no controle da expansão virtual da cobertura da PeNB conforme as condições de tráfego na célula.

Matematicamente, o esquema de seleção da célula por um usuário recebendo um tipo de serviço k , é definido da seguinte forma: seja $E = \{e_1, e_2, \dots, e_{|E|}\}$ o conjunto de estações rádio-base com valores de potência no sinal de referência $RSRP_u(e)$ medidas pelo terminal $u \in U = \{u_1, u_2, \dots, u_{|U|}\}$. Cada terminal u escolhe sua estação de serviço rádio com base na equação 6.1.

$$eNB(ID)/serviço_u = \underset{e}{\operatorname{argmax}} [RSRP_u(e) + CREO_k(e)] \quad (6.1)$$

$$CREO_k(e) = \begin{cases} CREO_k, & \text{se } e \text{ for uma PeNB} \\ 0, & \text{se } e \text{ for uma MeNB} \end{cases}$$

Note-se da equação 6.1 que o valor *offset* ($CREO_k$) é adicionado apenas para os valores de RSRP recebidos desde as PeNBs.

Nas seções seguintes, serão apresentadas tanto as métricas utilizadas para mensurar a qualidade percebida pelos usuários quanto a técnica cognitiva desenvolvida para a definição dinâmica dos *offsets* para cada tipo de serviço oferecido na rede.

6.2. Modelos de estimação da qualidade da experiência

O conceito de QoE está relacionado com a avaliação do desempenho das aplicações multimídia do ponto de vista dos usuários, colocando uma perspectiva pessoal de satisfação da informação transportada pela rede. Formalmente, a QoE é definida pela União Internacional de Telecomunicações (*International Telecommunication Union, ITU-T*) [81] como: "A aceitabilidade global de uma aplicação ou serviço, tal como é percebido subjetivamente pelo usuário final" (Tradução do autor)³.

Dentre as métricas mais comuns utilizadas para avaliar a QoE está a *Mean Opinion Score* (MOS). A MOS foi inicialmente proposta para quantificar a opinião subjetiva da qualidade de serviços de transmissão de voz e imagens. Tal opinião é dada por observadores que atribuem uma pontuação na escala de 1 a 5 conforme sua percepção de qualidade (Tabela 4).

Tabela 4. Escala de qualidade MOS

Qualidade	Valor da métrica	Grau do incômodo
Excelente	5	Imperceptível
Boa	4	Perceptível, mas sem incômodo
Regular	3	Leve incômodo
Pobre	2	Incômodo
Ruim	1	Muito incômodo

³ "The overall acceptability of an application or service, as perceived subjectively by the end-user" [81].

Dado que as avaliações subjetivas não são adequadas para seu uso em sistemas de otimização em tempo real, vários estudos têm desenvolvido modelos matemáticos para mapear e correlacionar as avaliações subjetivas com parâmetros de rede e QoS visando calcular a MOS de forma objetiva [82].

Neste capítulo, utilizamos um método de avaliação objetiva da MOS do tipo sem-referência [83]. Esta metodologia objetiva avaliar a qualidade do conteúdo multimídia recebido sem a necessidade de ter acesso ao conteúdo original transmitido, caracterizando a MOS como uma função dos parâmetros referentes ao tipo de serviço e métricas de desempenho de QoS.

A seguir são descritos os modelos empregados para estimar a MOS dos 3 tipos de serviços usados nos cenários de simulação: vídeo, voz e descarga de arquivos (dados).

6.2.1.

Modelo de avaliação de QoE para serviços de vídeo

Baseamos a predição da MOS dos serviços de vídeo a partir do modelo desenvolvido por Khan e colaboradores [84]. Este modelo estima a QoE por meio de uma combinação de parâmetros de aplicação e de rede, através da equação 6.2:

$$MOS_{\text{vídeo}} = \frac{a_1 + a_2 FR + a_3 \ln(SBR)}{1 + a_4 PER + a_5 (PER)^2}, \quad (6.2)$$

em que FR (*frame rate*) representa a taxa de quadros utilizada na codificação do vídeo, a SBR (*sender bit rate*) a taxa de codificação e a PER (*packet error rate*) a taxa de erro de pacotes.

Os autores calcularam os coeficientes a_v ($v = 1, 2, \dots, 5$) para três diferentes tipos de conteúdo de vídeo típicos: vídeo chamada (movimentos suaves), vídeo conferencia (movimentos moderados) e vídeo streaming (movimentos rápidos) (Tabela 5).

Tabela 5. Coeficientes da função MOS_video para as categorias de vídeo: movimentos suaves, moderados e rápidos

Coeficiente	Movimentos	Movimentos	Movimentos
-------------	------------	------------	------------

	suaves	moderados	rápidos
a_1	2.797	2.273	-0.0228
a_2	-0.0065	-0.0022	-0.0065
a_3	0.2498	0.3322	0.6582
a_4	2.2073	2.4984	10.0437
a_5	7.1773	-3.7433	0.6865

6.2.2.

Modelo de avaliação de QoE para serviços de voz

No caso da avaliação da qualidade dos serviços de voz sobre IP, adotamos o "modelo E" descrito no documento G.107 da ITU-T [85]. O objetivo deste modelo é determinar o fator de qualidade R (Fator-R). Este fator incorpora a análise de diversos parâmetros que influenciam na degradação da qualidade da fala nos sistemas de comunicações baseados em voz sobre IP. Após determinado o Fator-R, este pode ser mapeado para a escala subjetiva MOS.

Matematicamente, o cálculo do MOS é realizado a partir das equações 6.3 até 6.6:

$$MOS_{voz} = \begin{cases} 4.5, & R \geq 100 \\ 1 + 0.035R + R(R - 60) \times (100 - R) \times 10^{-6}, & R \in [0,100] \\ 1, & R \leq 0 \end{cases} \quad (6.3)$$

onde o Fator-R é determinado por:

$$R = R_0 - I_s - I_d - I_e + A \quad (6.4)$$

Na equação, R_0 representa a relação sinal-ruído básica, enquanto os fatores I_s , I_d , e A caracterizam as percepções do usuário. I_s simboliza a degradação que afeta diretamente o sinal de voz, tais como a intensidade da voz e o ruído de quantização.

O parâmetro I_d reflete a degradação causada por atrasos na rede. Sendo d a medida do atraso em um sentido da comunicação, o valor do I_d é calculado da seguinte forma:

$$I_d = \begin{cases} 0.024d, & d \leq 177.3ms \\ 0.024d + 0.11(d - 177.3), & d > 177.3ms \end{cases} \quad (6.5)$$

O parâmetro I_e representa as perdas relativas à tecnologia de transmissão tais como perda de pacotes na rede e *codecs* utilizados. Sendo P_{pl} a perda de pacotes em percentagem e B_{pl} o fator de robustez do *codec* em relação à perda de pacotes, o valor de I_e é definido por:

$$I_e = I_{e-c} + (95 - I_{e-c}) \frac{P_{pl}}{P_{pl} + B_{pl}}, \quad (6.6)$$

Os valores das variáveis I_{e-c} e B_{pl} dependem do tipo do *codec* utilizado. No caso do *codect* G.711, foram utilizados os seguintes valores: $I_{e-c} = 0$ e $B_{pl} = 4.3$ [85].

Finalmente, o parâmetro A é chamado de *fator de vantagem* e reflete o fato da qualidade percebida ser influenciada pela flexibilidade operacional para se estabelecer uma conexão. Este fator é tabelado para diferentes tipos de sistemas de comunicações, sendo utilizado o valor de 10 para sistemas móveis celulares.

6.2.3. Modelo de avaliação de QoE para serviços de transferência de arquivos

Para calcular a satisfação dos usuários com serviço de descarga de arquivos, utilizamos a relação definida em [86]. Este modelo caracteriza a MOS mediante uma função logarítmica dependente da taxa de recepção V_{rx} e a perda de pacotes P_e , tal como se expressa a seguir:

$$MOS_{dados} = \begin{cases} 4.5, & V_{rx} \geq V_{max} \\ a * \log_{10}[bV_{rx} + P_e], & V_{rx} \in [V_{min}, V_{max}] \\ 1, & V_{rx} \leq V_{min} \end{cases} \quad (6.7)$$

onde os parâmetros a e b são determinados a partir das taxas mínima (V_{min}) e máxima (V_{max}) equivalentes com as percepções de qualidade mínima (MOS=1) e máxima (MOS=4.5).

6.3. Mecanismo de aprendizagem autônomo para o balanceamento de carga em redes móveis multi-serviço

Nesta seção, descreve-se o desenvolvimento de um mecanismo cognitivo de balanceamento de carga entre macro e pico-células que é baseado na técnica CRE multi-serviço apresentada previamente na seção 6.1.

Especificamente, procurou-se que as PeNB aprendam a modificar dinamicamente sua cobertura conforme as condições de tráfego da célula, de tal forma que sejam consideradas as diferentes expectativas de QoE de usuários que utilizam diferentes tipos de serviços. Para isso, foi definido um mecanismo de aprendizagem, baseado na métrica MOS, como métrica comum entre as diferentes classes de serviços trafegados na rede de acesso.

A seguir apresentamos a descrição formal do problema de aprendizagem, os detalhes do algoritmo desenvolvido e os resultados obtidos através de processos de simulação.

6.3.1. Formulação do problema de aprendizagem

Como se ilustra na Figura 40, o cenário considerado consiste em uma rede móvel LTE heterogênea composta por estações rádio-base macro (MeNB) e pico (PeNB) que compartilham uma porção da sua área de cobertura.

Assim como no capítulo 5, consideramos as dificuldades na distribuição de usuários entre as macro e pico-células devido às diferenças de potência utilizadas pelos elementos na rede de acesso. Pontualmente, procuramos que as PeNB acrescentem dinamicamente sua área de cobertura, atraindo uma maior quantidade de UEs e, portanto, incrementando o nível de balanceamento de carga.

Neste capítulo, a diferença das técnicas desenvolvidas no capítulo 5, consideramos três tipos de aplicações nos cenários de simulação: vídeo, voz e transferência de arquivos. Além disso, os objetivos de aprendizagem e de otimização relacionados com a técnica de balanceamento de carga, centram-se na perspectiva dos usuários da célula, através da avaliação da métrica de QoE denominada MOS.

Em consequência, baseados na técnica de balanceamento de carga CRE-multi-serviço definida na seção 6.1, o problema é definido como:

Dado o conjunto $K = \{\text{vídeo}, \text{voz}, \text{dados}\}$ de classes de serviços, cada PeNB deve aprender a modificar de forma autônoma sua área de cobertura através do vetor de *offsets* $\overline{CREO} = [CREO_{\text{vídeo}}, CREO_{\text{voz}}, CREO_{\text{dados}}]$, em procura da otimização da QoE dos usuários na célula.

Neste capítulo, formulamos a solução de aprendizagem com base em uma abordagem de aprendizado por reforço livre de modelo. Desta forma, o agente

associado à PeNB aprende como ações específicas afetam o desempenho do sistema conforme seu estado atual. Este processo é realizado através de repetidas iterações de tentativa-erro com o ambiente e a recepção de sinais de recompensa/castigo.

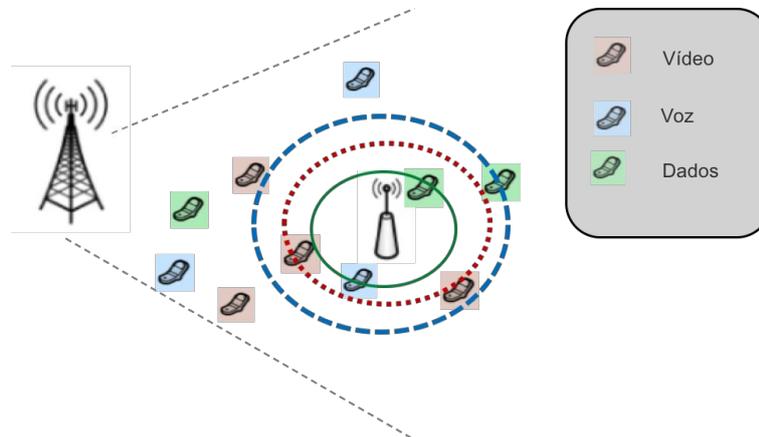


Figura 40. Expansão virtual de cobertura multi-serviço

Na sequência, apresentamos o processo de definição das variáveis de estado utilizadas na técnica desenvolvida para resolver o problema de balanceamento de carga multi-serviço.

6.3.2. Definição de variáveis de estado e análise dimensional

A definição das variáveis de estado é um fator essencial para caracterizar o estado do sistema de forma eficiente. Neste estudo, adotamos variáveis de estado relativas às métricas definidas nos padrões da tecnologia LTE listadas no documento 3GPP-TS 32.425 [87].

Algumas das variáveis adotadas foram triplicadas a fim de relacioná-las com cada tipo de serviço (vídeo, voz e dados) trafegado na rede. A seguir, descrevemos tais variáveis conforme o dispositivo de acesso que as percebe (MeNB ou PeNB):

6.3.2.1. Elementos percebidos pela MeNB

Dentre as variáveis percebidas pela MeNB utilizadas para descrever o estado da célula se encontram:

- $NRAB_{macro}^k$: variável que detalha o número de comunicações (*radio access bearers*) ativas com tráfego pertencente a classe de serviços k .
- ρ_{macro} : variável que representa a percentagem de utilização de blocos de recursos (recursos rádio).

A partir dos elementos descritos, o estado percebido pela macro-eNB é definido por:

$$S_{macro} = [NRAB_{macro}^{video}, NRAB_{macro}^{voz}, NRAB_{macro}^{dados}, \rho_{macro}] \quad (6.8)$$

As variáveis seleccionadas para ser percebidas pela MeNB procuram oferecer informações relativas à quantidade de recursos utilizados na macro-célula e à quantidade de comunicações oferecidas através deste dispositivo.

6.3.2.2. Elementos percebidos pela PeNB

Dentre as variáveis percebidas pela PeNB utilizadas para descrever o estado da célula se encontram:

- CQI_{5-pico}^k : variável que representa a qualidade do estado do canal dos usuários da borda da pico-célula com tráfego pertencente a classe de serviços k .
- ρ_{pico} : variável que descreve a percentagem de utilização de blocos de recursos na PeNB (recursos rádio).

$$S_{pico} = [CQI_{5-pico}^{video}, CQI_{5-pico}^{voz}, CQI_{5-pico}^{dados}, \rho_{pico}] \quad (6.9)$$

Em resumo, as variáveis escolhidas para ser percebidas pela PeNB refletem a possibilidade de expandir a cobertura da célula em função da qualidade atual dos usuários da borda da pico-célula e a totalidade de recursos utilizados por este dispositivo.

6.3.2.3. Análise dimensional

Nesta abordagem, a PeNB é elemento que toma a decisão sobre o valor do vetor de *offsets* *CREO*. A partir disto, realizamos inicialmente uma análise dimensional baseados na suposição de que a MeNB transmite todos seus elementos percebidos para a PeNB. Desta forma, a variável de estado de decisão é expressa pelo vetor de estado S dado por:

$$S = [NRAB_{Macro}^{video}, NRAB_{Macro}^{voz}, NRAB_{Macro}^{dados}, \rho_{Macro}, \rho_{Pico}, CQI_{5-Pico}^{video}, CQI_{5-Pico}^{voz}, CQI_{5-Pico}^{dados}] \quad (6.10)$$

Supondo uma discretização uniforme, em 3 intervalos, de cada elemento que compõe a variável de estado, e de cada variável de ação em 5 intervalos, o problema teria uma dimensão de $3^8 = 6561$ possíveis estados a serem explorados por um conjunto de $5^3 = 125$ possíveis ações.

A partir desta análise preliminar, identificamos que a introdução das variáveis utilizadas para descrever o desempenho de cada classe de serviço incrementou de forma exponencial o tamanho do espaço de estados/ações a ser considerado pelo algoritmo de aprendizagem. Tal incremento dificulta a utilização dos algoritmos tradicionais de aprendizagem por reforço.

Na subseção a seguir propomos uma modificação à estratégia de aprendizagem, adicionando uma estrutura hierárquica de subtarefas a fim de diminuir o impacto produzido pelo incremento das variáveis de estado. Com estratégia, procuramos representar e resolver o problema de balanceamento de carga multi-serviço, de tal forma que a PeNB aprenda a modificar o vetor de *offsets* \overline{CREO} (conforme o tráfego da rede) a fim de otimizar a QoE dos UEs da célula.

6.3.3. Solução de aprendizagem por reforço com estrutura hierárquica

Como mencionado na subseção anterior, o incremento do número de variáveis de estado/ação utilizadas pelo agente cognitivo, aumentam exponencialmente a complexidade da tarefa de aprendizagem. Para lidar com este desafio, também conhecido como a *maldição da dimensionalidade* (*Curse of*

dimensionality) [88]–[90], utilizamos uma metodologia fundamentada no princípio de "dividir para conquistar".

Seguindo essa metodologia, o problema original de aprendizagem é decomposto em subproblemas de menor complexidade que podem ser solucionados individualmente, mas tendo em conta a interação entre eles. Especificamente, desenvolvemos uma solução de aprendizagem por reforço através de um processo de decomposição hierárquica em dois níveis, os quais estão distribuídos entre os agentes associados à MeNB e à PeNB. Este processo permitiu criar abstrações espaciais e temporais no espaço de estados/ações do MDP original, resultando em uma série de subtarefas hierarquicamente organizadas (Figura 41).

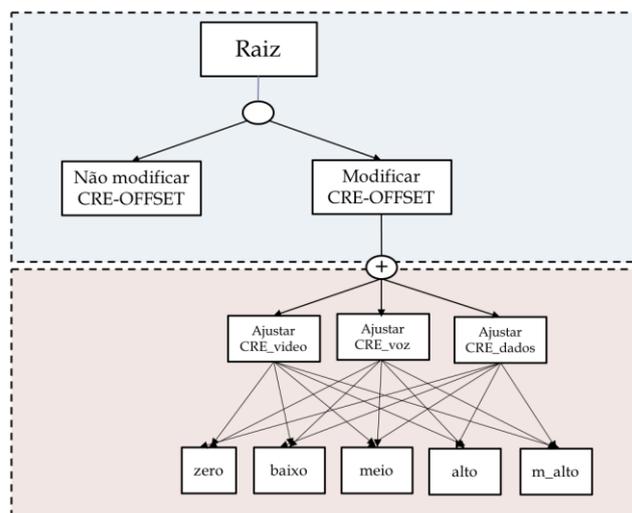


Figura 41. Estrutura hierárquica aplicada à técnica de expansão virtual de cobertura multi-serviço

Com o intuito de reduzir o número de estados em que o agente da PeNB vai precisar explorar durante a tarefa de aprendizado, é o agente associado a MeNB quem decide sobre a necessidade de realizar ou não o processo de balanceamento de carga conforme o estado geral da macro-célula, como observado no primeiro nível da Figura 41.

Este primeiro nível da estrutura hierárquica se baseia em uma decomposição do espaço de estados, sendo utilizada apenas a variável de carga da MeNB. Em caso que o agente associado à MeNB detecte uma situação de congestionamento, este solicita à PeNB a ativação do processo de expansão de cobertura (execução de macro ação “Modificar *CRE-OFFSET*”), iniciando o envio de informação de

estado/desempenho entre os dois agentes e dando início as tarefas do segundo nível da estrutura hierárquica. A política de ação para esta primeira subtarefa foi determinada de forma *a priori* a partir de resultados de simulação. Neste caso observou-se que para valores de ρ_{macro} inferiores a 0.85, a variação dos valores de *offset* não resultaram em melhoras significativas de desempenho para o sistema.

O segundo nível faz uso de abstrações temporais para descrever a tarefa CRE-multi-serviço como um conjunto de subtarefas organizadas e executadas sequencialmente na PeNB. Esta abstração foi realizada a partir da decomposição do vetor de políticas de macro ação “Modificar *CRE-OFFSET*” em subtarefas que determinem individualmente o valor de *offset* $CREO_k$ (política) que deve ser utilizado para cada tipo de serviço.

Do ponto de vista da tarefa de aprendizagem, em lugar de explorar os resultados para cada combinação de ação conjunta possível (combinação dos valores de *offsets* de cada classe de serviço), neste caso, apenas as políticas individuais de cada subtarefa devem ser aprendidas. Esta estratégia permitiu a decomposição do problema global em uma combinação de problemas de aprendizagem mais simples.

A Figura 42 apresenta a sequência temporal de subtarefas desencadeadas pela macro-ação “Modificar *CRE-OFFSET*” na PeNB. Este processo foi concebido como um mecanismo de decisão multi-objetivo sequencial, ordenado por prioridades e com base nos seguintes princípios: **i)** ajustar o $CREO_{video}$ otimizando a QoE para o grupo de UEs de vídeo, em seguida **ii)** ajustar o CRE_{voz} otimizando a QoE para o grupo de usuários com tráfego de voz desde que não sejam afetados os usuários de vídeo. Finalmente, **iii)** ajustar o CRE_{dados} otimizando a QoE dos usuários com tráfego de dados desde que não sejam afetados os usuários de voz nem os usuários de vídeo.

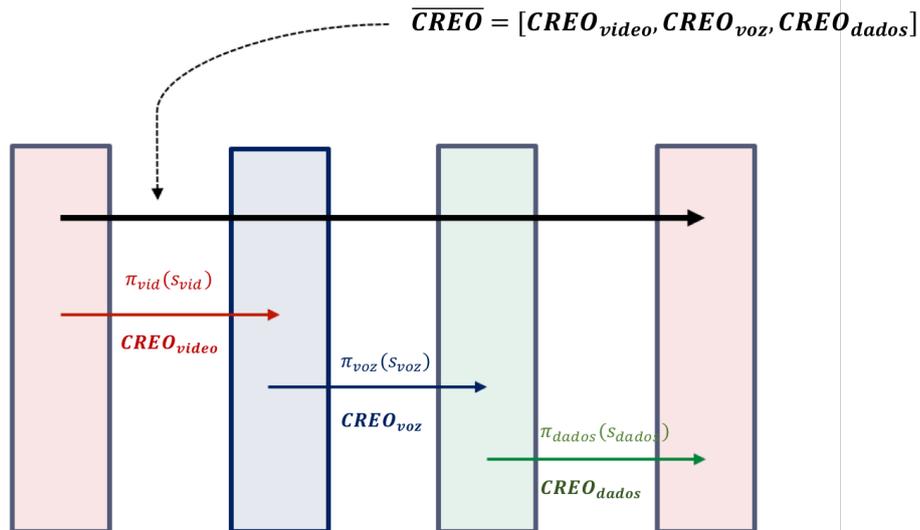


Figura 42. Sequência temporal de subtarefas desencadeadas pela macro-ação: Modificar CRE-OFFSET

A estratégia utilizada neste capítulo, baseada na decomposição de uma macro-ação em $|K| = 3$ subtarefas sequenciais, não permite garantir a otimalidade da ação global. Porém, dado que cada subtarefa aprende a melhor ação possível, o objetivo é que a interação entre as subtarefas e a combinação de suas políticas leve a um desempenho satisfatório (sub-ótimo) conforme as prioridades de otimização. Neste caso, estamos trocando otimalidade por simplicidade a fim de abordar e diminuir a complexidade da tarefa de aprendizagem global.

A seguir se descreve o processo de modelagem e o processo de aprendizado desenvolvido a partir da definição da estrutura hierárquica.

6.3.4. Modelagem e processo de aprendizado para cada subtarefa

Como resultado do processo de decomposição hierárquica, a técnica de balanceamento de tráfego multi-serviço é caracterizada pela decomposição do problema em subtarefas de menor complexidade. Estas subtarefas seguem uma estrutura de execução preestabelecida, porém, são os mecanismos de aprendizagem por reforço que levam ao agente a descobrir automaticamente as melhores políticas para a escolha das ações executadas por cada subtarefa.

Tal como mencionado previamente, a estrutura desenvolvida apresenta dois níveis, sendo que o processo de aprendizagem de políticas (valores de *offsets*)

ocorre no segundo nível da hierarquia. A seguir listamos as principais características na modelagem das subtarefas definidas no segundo nível da estrutura hierárquica:

- Cada subtarefa é descrita por um $subMDP_k$ ($k \in \{video, voz, dados\}$) com suas próprias abstrações no espaço de estados, ações e função de recompensa.
- Cada $subMDP_k$ é associado a uma função Q_k local que depende apenas de um subconjunto de todos os possíveis estados e ações.
- Cada $subMDP_k$ ativa-se apenas durante um período de tempo $\tau = T/3$.
- Os $subMDPs$ encontram-se acoplados entre eles pela variável de estado de carga da pico-célula.

Na sequência são descritos os componentes individuais (espaço de estados, ações e função de recompensa) de cada subtarefa definida na estrutura hierárquica.

6.3.4.1. Espaço de sub-estados

O espaço de estados para cada subtarefa é modelado a partir de um subconjunto da totalidade das variáveis de estado do problema. Neste caso foram utilizadas aquelas variáveis que descrevem as condições de tráfego da célula para cada tipo de serviço k , tal como se observa a seguir:

$$s_k = [NRAB_{macro}^k, \rho_{macro}^k, \rho_{pico}, CQI_{5-pico}^k] \quad (6.11)$$

É importante observar que a variável ρ_{pico} inclui-se em todos os espaços de estados independentemente da classe de serviço avaliado. Esta variável serve como acoplamento entre as subtarefas para indicar as mudanças de estado na PeNB uma vez realizada a mudança de cada $CREO_k$. Isto devido a que a execução de cada subtarefa modifica as condições de operação da célula, mudando o estado e os recursos rádio disponíveis na PeNB. Complementarmente ao descrito na seção 6.3.2, adicionamos em cada subtarefa a variável ρ_{macro}^k para descrever a percentagem de utilização de recursos do serviço k . Esta variável

permitiu incluir informações sobre a relação entre os tipos de serviços trafegados na MeNB, indicando o grau de ocupação do serviço k frente aos outros tipos de serviços.

A partir dessa segmentação das variáveis de estado, o sistema de inferência de cada subtarefa define-se por um número de 108 regras possíveis (resultado da combinação dos antecedentes). Estas regras serão exploradas conforme se apresentem durante a fase de treinamento do sistema (Seção 6.5.3).

6.3.4.2. Ações

Com base no estado avaliado, cada subtarefa determina como ação o valor do CRE *offset* a ser empregado pelo grupo de usuários com serviço k . Como notação, esta ação é indicada pela variável $CREO_k$.

6.3.4.3. Funções de recompensa

O sistema de recompensas utilizado foi definido a partir da estimação da QoE em termos da métrica MOS tal como apresentado na seção 6.2. Assim, a cada tempo τ a PeNB computa e armazena o desempenho para cada tipo de serviço no vetor:

$$MOS(t) = [MOS_{video}(t), MOS_{voz}(t), MOS_{dados}(t)] \quad (6.12)$$

Com base nesses valores, utilizamos o conceito de recompensas relativas para determinar o grau de efetividade da ação aplicada. A intenção é comparar as métricas obtidas em dois períodos de observação consecutivos através da variável vetorial dada por:

$$\begin{aligned} \Delta MOS(t) &= MOS(t) - MOS(t - \tau) \\ &= [\Delta MOS_{video}(t), \Delta MOS_{voz}(t), \Delta MOS_{dados}(t)] \end{aligned} \quad (6.13)$$

Desta forma, um valor de $\Delta MOS_k > 0$ sugere que a ação tomada foi favorável para o serviço k enquanto que se $\Delta MOS_k < 0$ a ação realizada foi prejudicial para este tipo de serviço.

O sistema de recompensas foi projetado através de um processo de escalarização. Isto permitiu transformar a natureza do problema multiobjetivo num problema mono-objetivo, facilitando a integração das técnicas tradicionais de aprendizagem por reforço [91]. Especificamente, implementamos as funções de recompensa de cada sub tarefa k como uma combinação linear determinada pela equação 6.14:

$$R_k(t) = w_k \Delta MOS(t)^T, \quad (6.14)$$

onde w_k é o vetor de pesos que representa a importância relativa de cada tipo de serviço em cada sub tarefa k . Com base nos critérios de otimização definidos na seção 6.3.3, os vetores de pesos foram definidos da seguinte forma:

$$w_{video} = \left[\frac{8}{10}, \frac{1}{10}, \frac{1}{10} \right] \quad (6.15)$$

$$w_{voz} = \left[\frac{4}{10}, \frac{4}{10}, \frac{2}{10} \right] \quad (6.16)$$

$$w_{dados} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right] \quad (6.17)$$

A Figura 43 resume os elementos de cada sub tarefa descritos nesta seção, sendo estes: o espaço de estados (s_k), ações ($CREO_k$) e função de recompensa (R).

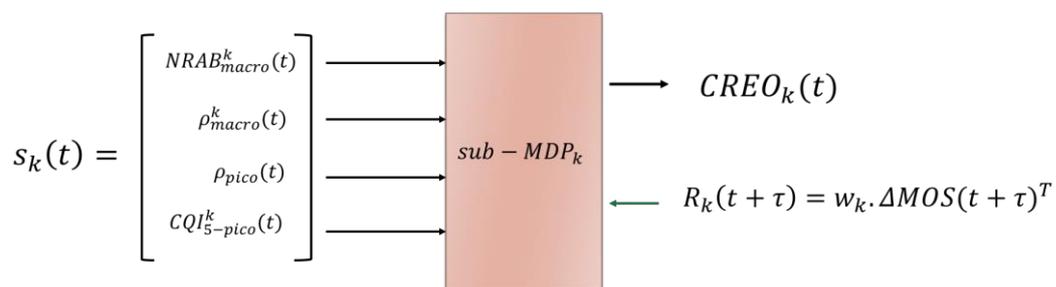


Figura 43. Definição dos elementos de cada sub-MDP

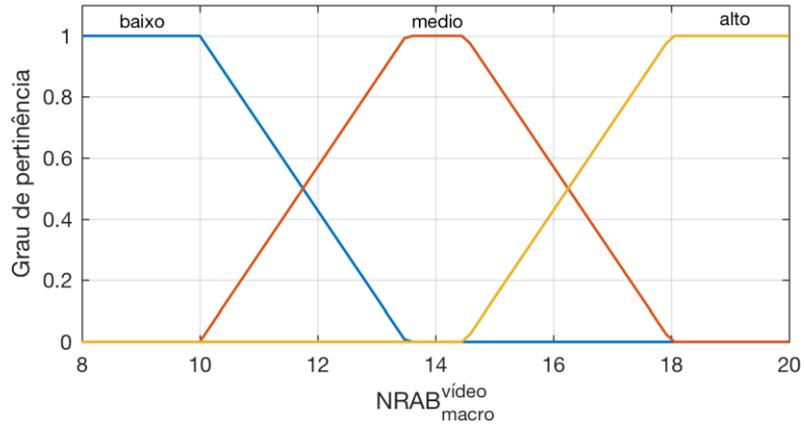
Na sequência, apresentamos os detalhes da implementação mecanismo de aprendizagem para as sub tarefas mencionadas nesta seção.

6.3.5. Implementação do mecanismo Fuzzy Q-Learning hierárquico

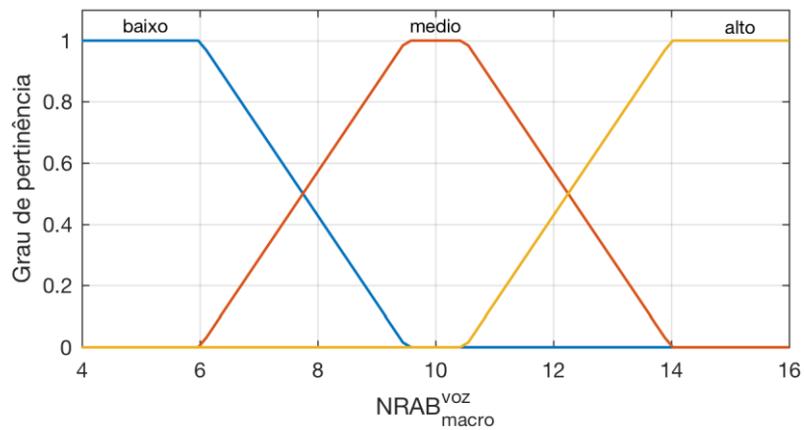
A partir da definição de sub-MDPs, o treinamento de cada subtarefa é realizado de forma individual, visando obter para cada uma delas uma função de política $\pi_k(s_k) \rightarrow CREO_k$ que mapeie os subestados em valores de *offsets*, maximizando a recompensa esperada. Esta etapa de treinamento foi realizada usando uma abordagem de aprendizagem por reforço sustentada na técnica *Fuzzy Q-learning* (livre de modelo). Lembrando da seção 5.4.1, o *Fuzzy Q-Learning* (FQL) é uma expansão da técnica de aprendizagem *q-learning* para seu uso em sistemas de inferência *Fuzzy* (FIS). Esta mistura de técnicas permitiu particionar as variáveis contínuas usadas para descrever o espaço de estados/ações em conjuntos *Fuzzy* finitos. Na sequência apresentamos a definição dos componentes do *FIS* em cada subtarefa assim como também os detalhes do algoritmo de aprendizagem por reforço implementado.

6.3.5.1. Definição dos FIS para cada subtarefa

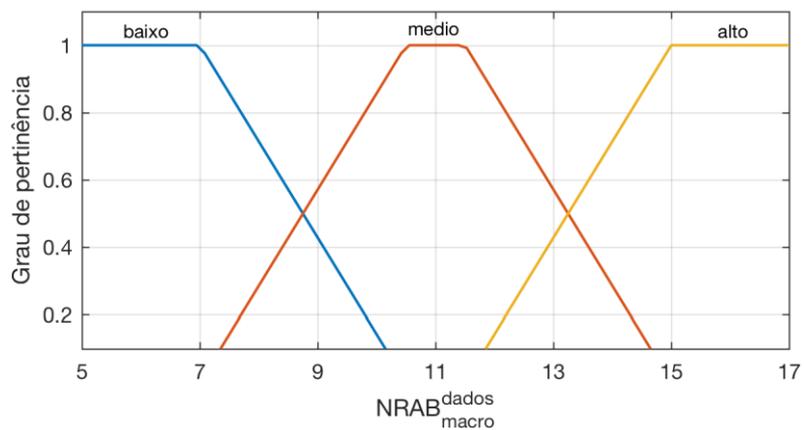
Na técnica multi-serviço, cada subtarefa k tem representado seu elemento de decisão por um FIS_k do tipo Takagi-Sugeno [92]. Desta forma, as variáveis de entrada de cada FIS se correspondem com os elementos das variáveis de estado s_k descritas na seção 6.3.4.1. Na solução adotada neste capítulo particionamos cada variável de estado em conjuntos *Fuzzy* através do uso de funções de pertinência com formato trapezoidal. Isto para evitar uma superposição excessiva entre os conjuntos, tal como aconteceria se fossem utilizadas funções triangulares. As Figuras abaixo ilustram a definição dos termos funções de pertinência para cada variável *Fuzzy* considerada na percepção do ambiente (estados). Estes conjuntos foram ajustados conforme o conjunto de dados utilizados na fase de treinamento (descritos na seção 6.5.1), envolvendo diversas intensidades de tráfego (estados) na rede.



a)



b)



c)

Figura 44. Funções de pertinência e conjuntos *Fuzzy* para a variável: número de comunicações (*radio access bearers*) ativas com tráfego pertencente à classe de serviços de a) vídeo, b) voz e c) dados

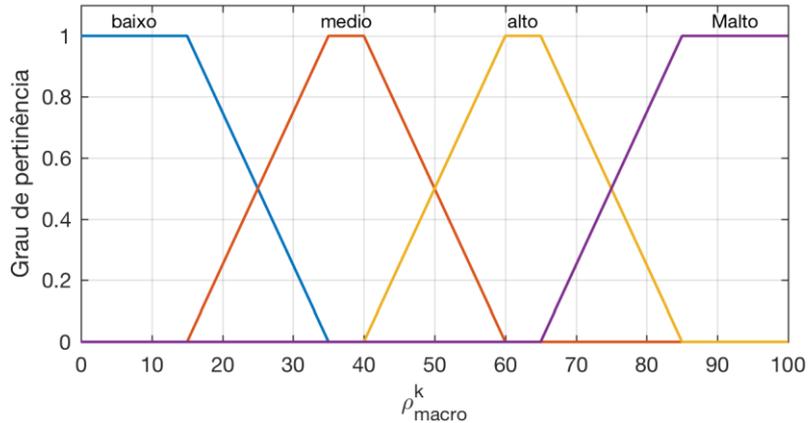


Figura 45. Funções de pertinência e conjuntos *Fuzzy* para a variável: percentagem de utilização de blocos de recursos na MeNB pela classe de serviços k

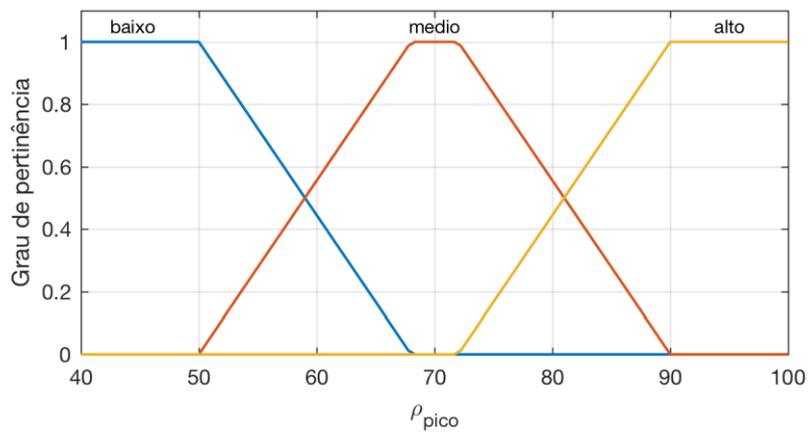


Figura 46. Funções de pertinência e conjuntos *Fuzzy* para a variável: percentagem de utilização de blocos de recursos na PeNB

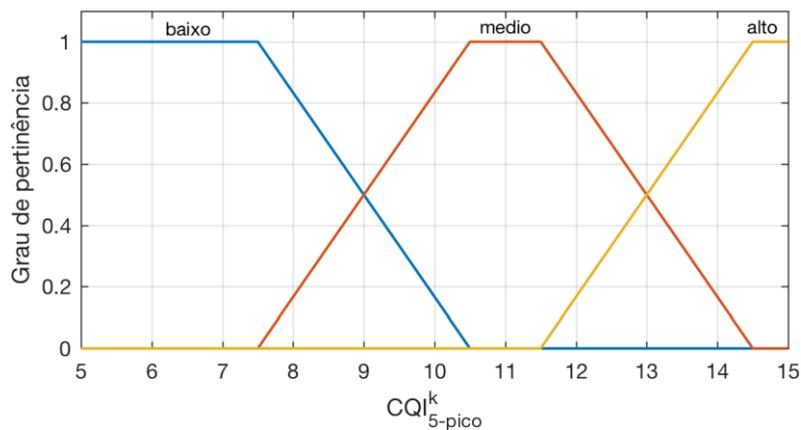


Figura 47. Funções de pertinência e conjuntos *Fuzzy* para a variável: qualidade do estado do canal dos usuários da borda da pico-célula com tráfego pertencente a classe de serviços k

A tomada de decisão em cada FIS é definido por um conjunto de regras do tipo "Se..., então..." formadas pela combinação das variáveis *Fuzzy* de entrada (antecedentes) e uma determinada ação a (consequente), neste caso relacionada com o valor do CRE -offset por tipo de serviço. Estes possíveis valores de *offsets* foram representados mediante funções de pertinência de saída com valores constantes ' o ' pertencentes ao conjunto:

$$O = \{0 \text{ (zero)}, 2 \text{ (baixo)}, 4 \text{ (medio)}, 6 \text{ (alto)}, 8 \text{ (muito alto)}\} \quad (6.18)$$

Em resumo, a execução dos FIS_k envolve 3 etapas: fuzzificação das variáveis de entrada s_k , a realização do processo de inferência e a defuzzificação da saída para a obtenção de um valor numérico de $CREO_k$. Na primeira etapa de fuzzificação, os valores numéricos de cada variável são transformados em graus de pertinência para cada um dos conjuntos *Fuzzy*. Na segunda etapa, o mecanismo de inferência identifica as regras ativadas pelos valores de entrada. Finalmente, na etapa de defuzzificação, calcula-se a saída numérica (ação) em função das regras ativadas e seus respectivos graus de ativação.

Neste caso, determinamos o valor numérico de saída ($CREO_k$) pela média ponderada dada pela equação 6.19,

$$CREO_k(s_k) = \frac{\sum_{i=1}^{N_k} \alpha_{i,k}(s_k) * o_i}{\sum_{i=1}^{N_k} \alpha_{i,k}(s_k)}, \quad (6.19)$$

onde N_k é o número de regras no FIS_k , $\alpha_{i,k}(s_k)$ é o grau de ativação da regra i para a entrada s_k e o_i é o consequente selecionado para dita regra.

6.3.5.2. Algoritmo FQL multi-serviço

Dado que inicialmente cada controlador *Fuzzy* começa seu funcionamento sem informação *a priori* do problema, o objetivo do algoritmo FQL é encontrar o melhor consequente para cada regra dos FIS_k (através de um processo de tentativa-erro com o ambiente) durante o processo de aprendizagem. Para isto,

cada regra $r_{n,k}$ ($n = 1, 2, \dots, N$) é associada com um valor de qualidade $q_k[n_k, o_j]$ ($j = 1, 2, \dots, J$) para cada possível consequente (ação/offset $o \in O$). A equação 6.20, mostra a composição das regras resultantes no controlador Fuzzy de cada subtarefa k .

$$\begin{aligned} \mathbf{Se} \{s_k = s_{n,k}\} \mathbf{ent\~{a}o}: \text{a\~{c}ao} = o_1 \text{ com } q_k[n, o_1] \mathbf{ou} \text{a\~{c}ao} \\ = o_j \text{ com } q_k[n, o_2] \mathbf{ou} \dots \mathbf{ou} \text{a\~{c}ao} \\ = o_j \text{ com } q_k[n, o_j] \end{aligned} \quad (6.20)$$

Desta forma, o elemento decisão (função q_k) é representado por uma matriz de dimensões $N_k \times J$, onde o número da linha representa o índice da regra do FIS_k e o número de coluna indica o índice da possível ação que será realizada (consequente).

Na etapa de **otimização**, em cada regra ativada, o consequente com maior valor de qualidade será escolhido para realizar cálculo do valor do *offset* em cada FIS_k (equação 6.21).

$$o_i = \underset{j}{\operatorname{argmax}} [q_k(n, o_j)] \quad (6.21)$$

De forma geral, o algoritmo de aprendizado explora sequencialmente a relação de pares antecedentes/consequentes (estado/ação) de cada FIS_k para determinar as políticas que maximizam suas recompensas. Para isto, em cada subtarefa k se realiza um ciclo iterativo descrito pelos seguintes passos:

- 1) Perceber o estado $s_k(t)$ do ambiente.
- 2) Para cada regra ativada no FIS_k escolher um consequente (explorar/otimizar), calcular o $CREO_k$ resultante e aplicá-lo no ambiente.
- 3) Observar o novo estado $s(t + \tau)$ do ambiente e a recompensa $R_k(t + \tau)$ recebida.
- 4) Aprender da experiência realizada (transição), atualizando a função Q_k

Nesta técnica multi-serviço, o processo de exploração (passo 2, equação 6.22) foi implementado utilizando a estratégia “*decaying e-greedy*”. Para isto,

através da equação 6.23 foi definida a probabilidade de exploração ϵ de cada subtarefa k como uma função decrescente, definida por partes e com limites estabelecidos entre $\epsilon_0 = 1$ e $\epsilon_f = 0.05$.

$$o_{n,k} = \begin{cases} \underset{j}{\operatorname{argmax}}[q_k(n, o_j)], & \text{com probabilidade } 1 - \epsilon(r_{n,k}) \\ j \text{ aleatorio,} & \text{com probabilidade } \epsilon(r_{n,k}) \end{cases} \quad (6.22)$$

$$\epsilon(r_{n,k}) = \min \left\{ \max \left[\epsilon_0 \cdot (0.5)^{\left\lfloor \frac{A(r_{n,k}) - \delta_k}{10} \right\rfloor}, \epsilon_f \right], \epsilon_0 \right\}, \quad (6.23)$$

$$\delta_k = \begin{cases} 0, & k = \text{vídeo} \\ 10, & k = \text{voz} \\ 15, & k = \text{dados} \end{cases}$$

onde $A(r_{n,k})$ é o grau de ativação acumulado da regra $r_{n,k}$ e δ_k é um fator de deslocamento conforme o tipo de serviço explorado. Na equação 6.23, $\lfloor \cdot \rfloor$ indica matematicamente o maior inteiro menor ou igual ao argumento (*floor*).

Esta estratégia permitiu modificar a probabilidade de exploração no tempo, conforme o tipo de serviço e o número de vezes (grau de ativação acumulado) que cada regra tiver sido explorada. Desta forma, conseguimos incentivar uma maior exploração ($\epsilon_0 = 1$) no início do período de treinamento, garantindo que todas as possíveis ações (consequentes) sejam testadas para cada uma das regras ativadas. Na medida em que as regras são exploradas, a taxa de exploração de cada regra decresce até um valor final ($\epsilon_f = 0.05$) em função do seu grau de ativação acumulado. Na Figura 48 se observa a evolução dos valores de exploração para os 3 tipos de serviço implementados. Estes valores foram definidos de forma diferenciada para estimular/incentivar um tempo de exploração maior nos serviços de voz e dados.

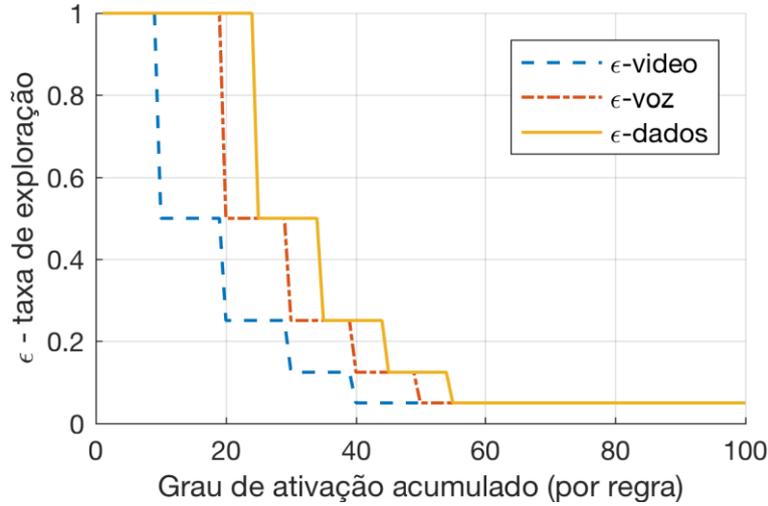


Figura 48. Funções de exploração

Uma vez determinados os consequentes a ser aplicados pelas regras ativadas para o estado s_k atual, o FIS_k calcula o $CREO_k$ (equação 6.19) e seu valor- Q_k global (equação 6.24).

$$Q_k(s_k, CREO_k) = \frac{\sum_{i=0}^{N_k} \alpha_{i,k}(s_k) q_k[i, o_i]}{\sum_{i=1}^{N_k} \alpha_{i,k}(s_k)} \quad (6.24)$$

O $CREO_k$ resultante é ajustado na rede. Após disso, se observa a evolução do sistema para o estado $s(t + \tau)$ e a recompensa $R_k(t + \tau)$ obtida. A partir dessa informação os valores da matriz q_k são atualizados através de um processo de diferenças temporais. Este processo de atualização é similar ao usado em *q-learning*, com a diferença que a taxa de aprendizado λ é ajustada segundo o grau de ativação de cada regra $r_{n,k}$ ativada, tal como se observa na equação 6.25

$$q_k[n, o_i] = q_k[n, o_i] + \lambda \cdot \left[\frac{\alpha_{n,k}(s_k)}{\sum_{i=1}^N \alpha_{i,k}(s_k)} \right] \cdot \Delta Q_k, \quad (6.25)$$

onde ΔQ_k é a diferença entre $Q_k(s_k(t + \tau), CREO_k)$ e $Q_k(s_k(t), CREO_k)$ calculada pela equação 6.26

$$\Delta Q_k = [R_k(t + \tau) + \gamma \cdot V(s(t + \tau))] - Q_k(s_k(t), CREO_k), \quad (6.26)$$

Sendo γ é o fator de desconto e $V(s(t + \tau))$ representando o valor do próximo estado em $t + \tau$ (equação 6.27):

$$V[s(t + \tau)] = \frac{\sum_{i=0}^{N_k} \alpha_{i,k}(s_k) \cdot \max_j \{q_k[i, o_j]\}}{\sum_{i=1}^{N_k} \alpha_{i,k}(s_k)} \quad (6.27)$$

Durante o processo de aprendizagem o fator de desconto γ teve um valor de 0.3, indicando uma maior relevância na recompensa imediata $R_k(t + \tau)$.

O algoritmo 2 resume o processo de aprendizagem multi-serviço.

Algorithm 2 Algoritmo Fuzzy Q-Learning

- 1 Observar o estado da MeNB $s_0(t) = [\rho_{macro}]$
 - 2 **Se** $\rho_{macro} < L$ **Então**
 - 3 Não ajustar o \overline{CREO}
 - 4 $t \leftarrow t + 1$
 - 5 **Senão**
 - 6 **para** $k \leftarrow$ vídeo **até** dados **faça** \triangleright para cada tipo de serviço
 - 7 Observar o vetor de estado $s_k(t)$
 - 8 Para cada regra $r_{n,k}$ ativada selecionar um conseqüente $o_{n,k} \in O$ conforme a política de exploração/otimização (**equações 6.22 e 6.23**)
 - 9 Calcular o $CREO_k$ global inferido pelo sistema de controle fuzzy e seu correspondente valor-q global: $Q_k(s_k(t), CREO_k)$ (**equações 6.19 e 6.24**)
 - 10 Realizar a ação indicada pelo controlador $Fuzzy$ e deixar o sistema evoluir ao próximo estado $s(t + \tau)$
 - 11 Observar o sinal de reforço (recompensa) $R_k(t + \tau)$
 - 12 Atualizar a matriz de valores- q_k através de diferenças temporais (**equação 6.25**)
 - 13 $t \leftarrow t + \tau$
 - 14 **fim para**
 - 15 **fim Se**
-

6.4. Implementação na arquitetura LTE

A integração da técnica de balanceamento de carga cognitivo foi realizada com base nos conceitos definidos para a arquitetura CCSON [31] descritos no capítulo 4. Especificamente, foi implementado um grupo cognitivo composto por um agente cooperativo associado a MeNB e um agente cognitivo associado a PeNB. Estes agentes compartilham entre eles informações de estado e

desempenho através da interface X2 conforme seja requerido pela técnica de balanceamento de carga e o estado da célula. As Figura 49 e Figura 50 ilustram os componentes funcionais dos agentes envolvidos neste processo.

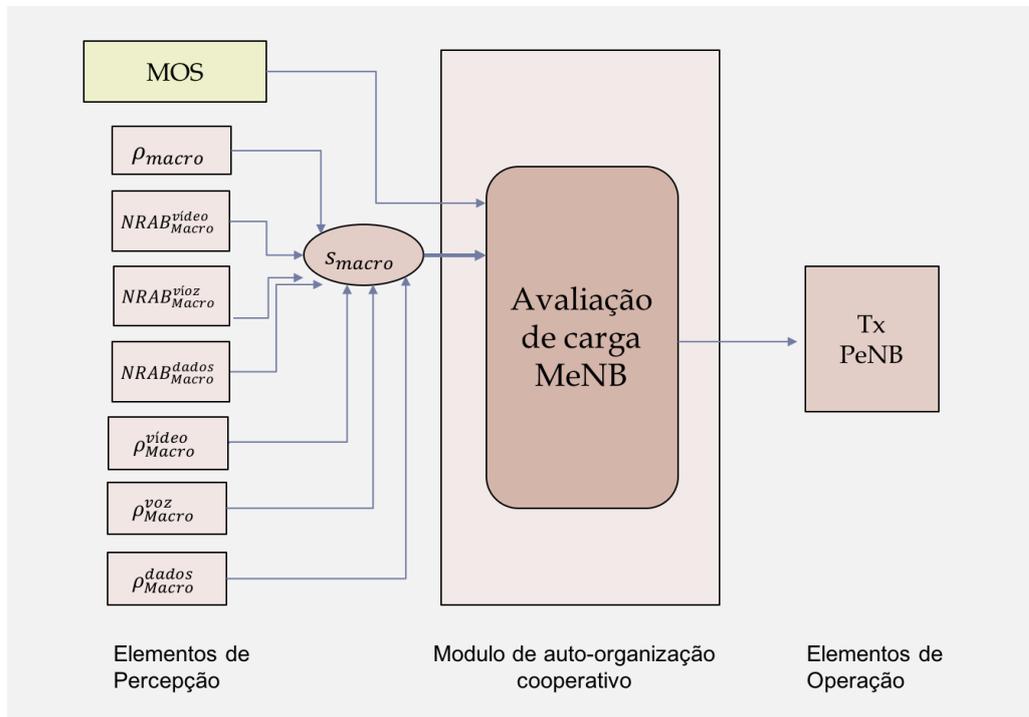


Figura 49. Agente cooperativo associado a MeNB

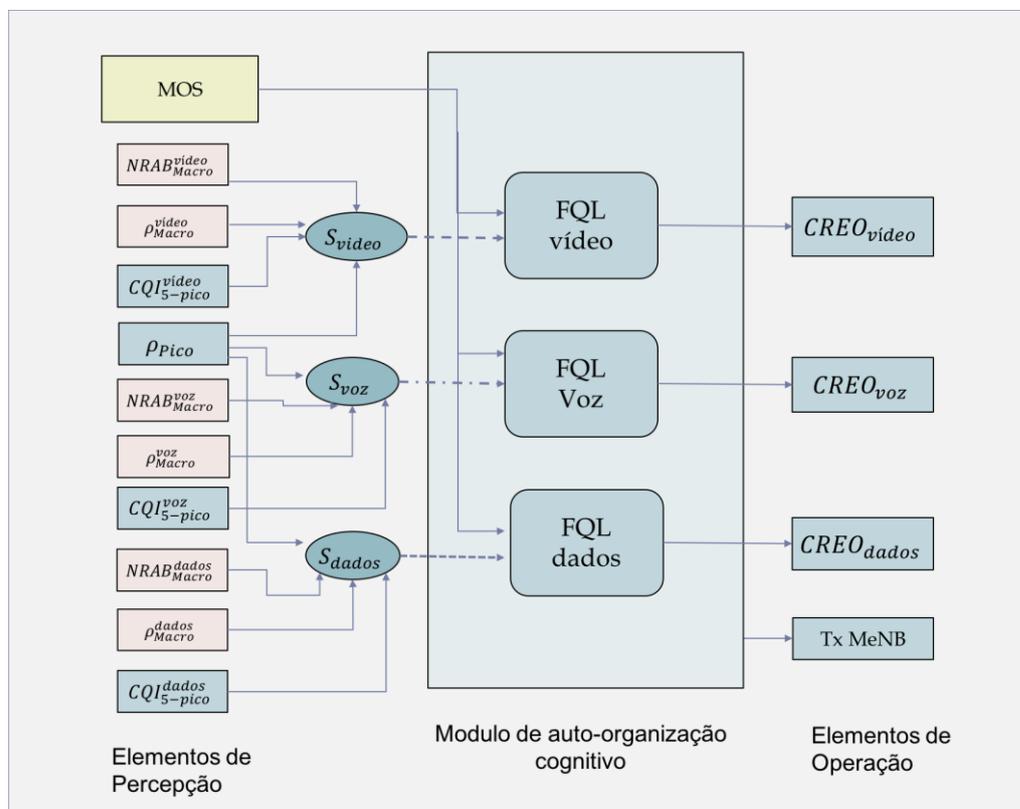


Figura 50. Agente cognitivo associado a PeNB.

Tal como mencionado na seção 6.3.3, inicialmente a MeNB detecta a situação de congestionamento e solicita a ativação do processo de expansão de cobertura (além do envio periódico de informação de estado/desempenho) para a PeNB em procura de aliviar sua carga de tráfego. A partir desse momento, o agente associado na PeNB realiza a tomada de decisão sobre os valores de *offsets* a serem informados para os usuários com base nas informações fornecidas pela MeNB e as medidas de desempenho/estado locais. A Tabela 6 resume os principais aspectos da técnica de balanceamento de carga multi-serviço desenvolvido.

Tabela 6. Resumo dos aspectos principais da técnica de balanceamento cognitivo multi-serviço

Objetivo	CRE-multi-serviço: Modificar os valores de <i>offsets</i> (para cada classe de serviço) que incrementam a cobertura da PeNB em procura da melhora da QoE dos usuários da célula.
Elemento decisor	Pico eNB
Dados de entrada	Informações de estado da célula: estatísticas de operação e medidas de desempenho.
Mecanismo de cooperação	Formação de um grupo cognitivo entre a MeNB e a PeNB apoiado pela troca de mensagens através da interface X2.
Algoritmo de aprendizado	Fuzzy Q-Learning + Definição de uma estrutura hierárquica de subtarefas.

6.5.

Descrição do cenário simulação e a etapa de aprendizagem

Nesta seção apresentamos inicialmente a descrição do cenário de simulação utilizado para avaliar a técnica de balanceamento, a modelagem utilizada para a simulação de cada tipo de serviço e processo detalhado para a simulação da etapa de aprendizagem.

6.5.1.

Descrição do cenário de simulação

Para a realização das experiências de treinamento e avaliação de desempenho, consideramos um cenário LTE com 7 macro-sites separados por 500 metros. Cada site é composto por três setores (3 eNBs) e uma pico-eNB

omnidirecional localizada na borda da área de cobertura tal como se ilustra na Figura 51.

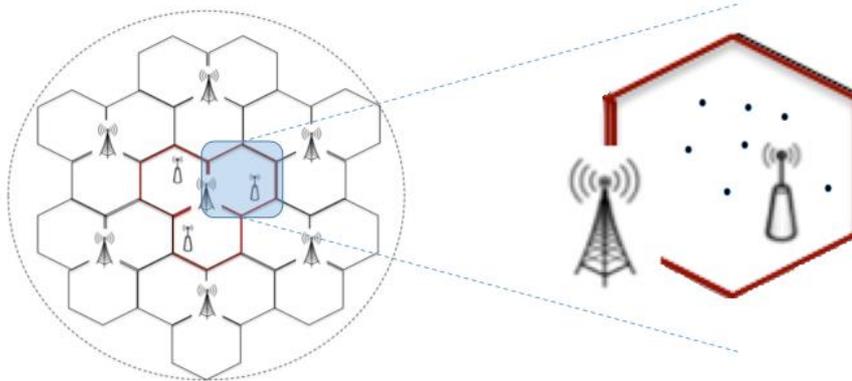


Figura 51. Cenário de simulação

Neste caso de estudo, o centro da célula representa a área de interesse enquanto o resto das estações rádio-base geram interferência aos UEs examinados. Os UEs são uniformemente distribuídos na área de cobertura avaliada e configurados com um padrão de baixa mobilidade.

Nos cenários simulados, consideramos $|k| = 3$ tipos de serviços: vídeo, voz e descarga de arquivos (dados). Assumindo que cada UE solicita apenas um dos tipos de serviços disponíveis e fundamentados nas previsões de tráfego para redes móveis pela Cisco [93], estabelecemos a distribuição de tráfego através da função de probabilidade discreta enunciada na Tabela 7.

Tabela 7. Função de probabilidade para a distribuição de tráfego nos cenários de treinamento

k	Vídeo	Voz	Dados
$P(k)$	0.45	0.2	0.35

A Tabela 8 mostra os principais parâmetros e modelos de propagação utilizados na rede de acesso (*Evolved Universal Terrestrial Radio Access Network, E-UTRAN*). Estes valores estiveram baseados nos parâmetros definidos pela 3GPP para a simulação de sistemas heterogêneos, especificamente para o modelo 1 (*Urban Macro cell + outdoor Hotzone cells*, tabela A.2.1.2-3 em [72]).

Tabela 8. Parâmetros de simulação do E-UTRAN [72]

<i>Parâmetro</i>	<i>Valor</i>
Frequência da portadora	2 GHz
Largura de banda	5 MHz
Distância entre as MeNB (<i>ISD</i>)	500 m
Potência de transmissão MeNB	46 dBm
Potência de transmissão PeNB	30 dBm
Algoritmo de escalonamento de pacotes	<i>Proportional Fair</i>
Perdas de propagação - MeNB	$128.1 + 37.6 \cdot \log_{10} R$
Perdas de propagação - MeNB	$140.7 + 36.7 \cdot \log_{10} R$
Perdas de penetração	20 dB

6.5.2.**Modelagem e simulação dos tipos de serviços trafegados na célula**

O NS-3 permite criar abstrações de aplicações de transmissão de dados e emular seu comportamento sobre diversos tipos de redes de comunicações (reais ou simuladas). Estas aplicações compõem um modelo cliente/servidor para a geração de pacotes interagindo diretamente com as camadas próprias da tecnologia de transmissão utilizada.

A simulação do tráfego de vídeo foi realizada mediante a utilização da ferramenta Evalvid [94] e sua integração no simulador NS-3 através de um módulo desenvolvido pelo grupo de estudo em redes de computadores na universidade Federal do Pará [95]. A ferramenta Evalvid permite obter a partir de um vídeo real, um arquivo de referência com todos os detalhes dos pacotes a ser transmitidos na rede simulada (tempos de transmissão e tamanho em bytes). Nos cenários implementados, utilizamos o vídeo "*carphone*" [96], codificado no padrão H.264/AVC com uma taxa de 512kbps.

A transmissão de serviços de voz foi modelada a partir de serviços conversacionais VoIP com supressão de silêncio e codec G.711 (64 kbit/s). Para isto, utilizamos uma aplicação do tipo *ON-OFF*, emulando os períodos de fala (transmissão de dados) e silêncio (sem transmissão). Estes tempos de fala e silêncio foram definidos aleatoriamente através de distribuições exponenciais com médias de 0.352 e 0.650 segundos, respectivamente.

Finalmente, a simulação do serviço de dados (transferência de arquivos) foi realizada por uma aplicação TCP configurada para solicitar a transferência de pacotes com uma taxa máxima de transmissão de 800kpbs.

6.5.3. Simulação do processo de aprendizagem

Tal como mencionado no Capítulo 4, o cenário proposto foi simulado através do simulador de código aberto NS-3, o qual foi integrado com Matlab (ferramenta utilizada para implementar o algoritmo de aprendizado automático). Este algoritmo modifica dinamicamente os *offsets* que determinam a cobertura da pico-célula para cada tipo de serviço.

Na etapa de aprendizagem foram considerados 1000 episódios de treinamento visando simular diversas intensidades de tráfego (estados) na rede. Para isto, em cada um dos episódios variamos o número de UEs de forma aleatória no intervalo [35,50] usando a distribuição de serviços mencionada na seção 6.5.1.

No início da etapa de aprendizagem todos os valores das tabelas Q foram inicializadas em zero. Isto para representar que a PeNB não possui informação *a priori* sobre as políticas de ações do problema (ou em outras palavras, as informações sobre como o agente deve agir para maximizar o desempenho do sistema).

A Figura 52 ilustra o processo detalhado durante cada episódio de treinamento. Especificamente, cada episódio abrange 7 períodos sequenciais de simulação equivalentes a 21 segundos de tempo real de operação da rede. No final de cada período a PeNB obtém as estatísticas relativas ao estado da célula para posteriormente realizar o processo de exploração/otimização do sistema através da mudança dos *offsets* para cada tipo de serviço. Conjuntamente, são calculadas as estatísticas de desempenho que determinam o valor da recompensa obtida pela ação realizada permitindo a atualização das tabelas Q de cada tipo de serviço.

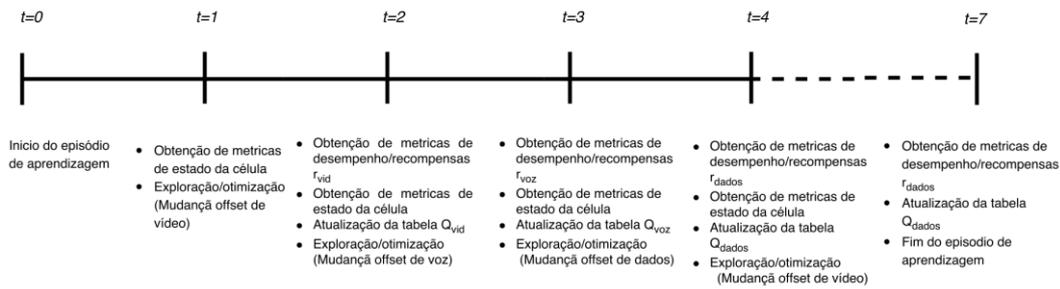


Figura 52. Descrição do processo em cada episódio de treinamento durante a fase de aprendizagem

6.6. Avaliação de desempenho

Nesta seção, inicialmente apresentamos os resultados obtidos durante a etapa de aprendizagem, através de uma análise da evolução temporal no funcionamento do algoritmo proposto. Seguidamente, apresentamos uma análise de desempenho do algoritmo na etapa de otimização, realizando comparações com outras abordagens.

6.6.1. Evolução temporal durante a fase da aprendizagem

Durante a fase de aprendizagem a PeNB explora diversos *offsets* conforme o estado de tráfego da célula, atualizando constantemente as tabelas Q (elementos de decisão). Com o objetivo de analisar a convergência do sistema, foi definido um cenário de observação no qual fosse possível verificar a evolução temporal do funcionamento do algoritmo durante a etapa de aprendizagem. Este cenário é composto por 45 UEs (representando uma situação de congestionamento), entre os quais 18 UEs recebem uma comunicação de vídeo, 11 UEs uma comunicação de voz e 16 UEs uma descarga de arquivos. No total, foram realizadas 100 observações espalhadas aleatoriamente no processo de treinamento. Os resultados destas observações se ilustram na Figura 53 e Figura 54. Especificamente, a Figura 53 mostra a evolução dos valores $CREO_k$ a cada iteração enquanto a Figura 54 apresenta a QoE resultante para cada tipo de serviço. Nestas Figuras, é possível observar como o algoritmo FQL (de cada subtarefa) explora amplamente os valores de *offsets*, gerando diversos desempenhos de qualidade na célula. Na

medida em que o número de iterações aumenta, os *offsets* convergem para aqueles valores que produzem os resultados de QoE conforme o expressado nas preferências de otimização. Neste caso, priorizando os serviços de vídeo frente aos serviços de voz e dados. Da Figura 53, também consegue-se observar como a sub tarefa responsável por modificar o $CREO_{vídeo}$ alcança a estabilidade antes que as subtarefas que modificam o $CREO_{voz}$ e $CREO_{dados}$. Isto se deve a forma na qual foi definida a função de exploração para cada tipo de serviço (equação 6.22), e ao acoplamento existente entre as subtarefas sequencias.

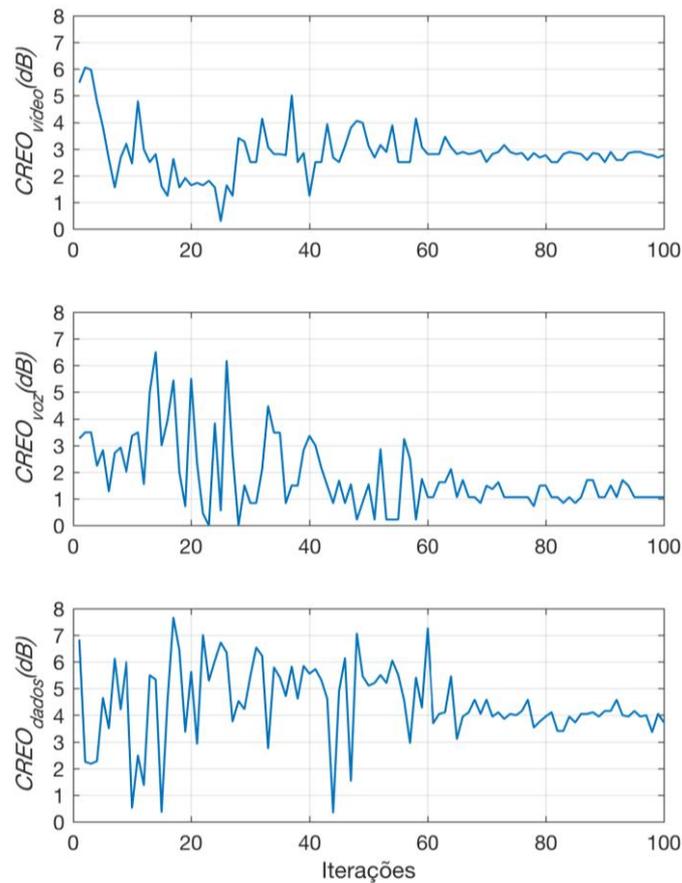


Figura 53. Evolução temporal dos valores de *CRE-offsets* de vídeo, voz e dados

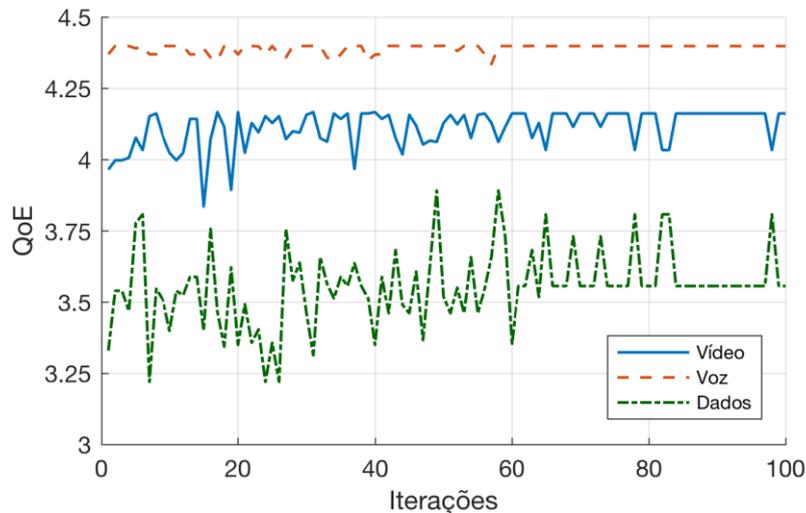


Figura 54. Evolução temporal da QoE média por classe de serviço como consequência na mudança dos *CRE-offsets* (vídeo, voz e dados)

Após o período de treinamento, as tabelas q_k de cada subtarifa contêm as informações para determinar as melhores ações (consequentes) no conjunto de regras exploradas em cada FIS_k . Com isto, na próxima seção apresentaremos o desempenho do sistema resultante em cenários de teste para diferentes intensidades de tráfego.

6.6.2. Avaliação de desempenho na etapa de otimização

Uma vez terminados os episódios de treinamento, realizamos a avaliação de desempenho da técnica resultante (CREO-Multi-serviço) com objetivo de observar a efetividade da técnica de aprendizagem proposta. Os testes foram realizados para várias condições de tráfego na célula, calculando os ganhos na vazão total (pelo incremento na cobertura da PeNB) e a QoE resultante para os tipos de serviço estudados.

A Tabela 9 apresenta as características do tráfego implementados nos cenários de teste. Os resultados foram calculados a partir de 20 simulações para cada quantidade de UEs considerados na rede.

Tabela 9. Condições de tráfego utilizadas nos cenários de teste

Parâmetro de tráfego na célula	Valor
Número total de UEs ativos	[35, 40, 45, 50]
Percentagem de UEs de vídeo	45%
Percentagem de UEs de voz	20%
Percentagem de UEs de dados	35%

A avaliação de desempenho da técnica proposta neste capítulo foi comparada com *i)* a técnica tradicional de seleção de célula baseada nos níveis de RSRP (MaxRSRP) e *ii)* o valor ótimo do *CRE-Offset* que maximiza a vazão total na célula (MaxTh).

Dado que a técnica ABC [77], utilizada como técnica de comparação no capítulo 5, não é adaptável para o cenário multi-serviço, desenvolvemos um mecanismo de busca exaustiva para determinar o *offset* que resultasse na máxima vazão da célula (emulando o objetivo da técnica ABC). Este processo consistiu em realizar para cada cenário de teste, simulações variando os valores do *CREO* até encontrar o valor que maximiza a vazão na célula. Isto permitiu comparar as vantagens/desvantagens entre a técnica proposta (CREO-Multiserviço) orientada à otimização de métricas de QoE percebida pelos UEs e uma abordagem tradicional baseada na otimização de métricas de rede (aqui chamada de MaxTh).

A Figura 55 ilustra a relação entre a vazão total da célula (MeNB+PeNB) e número de UEs ativos. Da Figura, observa-se a efetividade da técnica CRE para incrementar o desempenho da rede. Isto, devido a que ao incrementar virtualmente a cobertura da pico-célula, um número maior de UEs conseguem se associar com a PeNB, reduzindo o congestionamento na MeNB e consequentemente melhorando a vazão global.

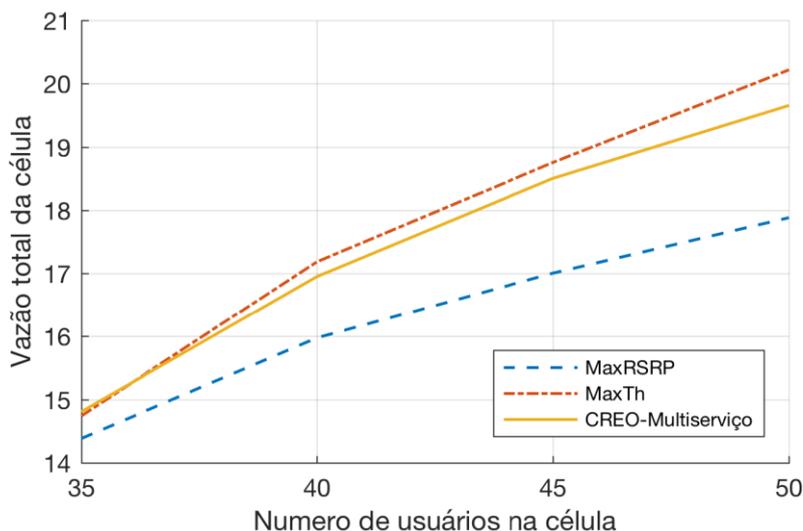


Figura 55. Vazão total da célula vs. número de UEs ativos

Neste caso, observamos que a técnica MaxTh apresenta melhores resultados em termos da vazão global da célula, porém muito próximos da técnica CREO-Multi-serviço proposta nesta tese. Neste caso de estudo, MaxTh indica que através da técnica *CRE* é possível conseguir um aumento de 1.7Mbps (10.28%) e 2.3Mbps (13.06%) na vazão global da célula para uma quantidade de 45 e 50 UEs respectivamente, quando comparado com a técnica tradicional MaxRSRP. Na mesma comparação, a técnica proposta CREO-multi-serviço obteve valores de aumento de 1.49 Mbps (8.81%) e 1.77 Mbps (9.91%).

Focando a atenção na análise de QoE, a Figura 56 mostra a degradação da QoE média na célula quando o número de UEs aumenta. Neste caso, observa-se como os ganhos na percepção da qualidade nos UEs pelo uso da técnica CRE são significativos na medida que a carga da rede aumenta. Observa-se também que embora a técnica CRE-multi-serviço não obtivesse os resultados ótimos em termos da vazão global da célula, esta técnica (baseada nas MOS e centrada na qualidade percebida pelos UEs) consegue melhorar a QoE média global na célula. Este resultado confirma o concluído por abordagens na literatura que mencionam as vantagens do desenvolvimento de mecanismos de otimização centrados na perspectiva dos UEs [97][98].

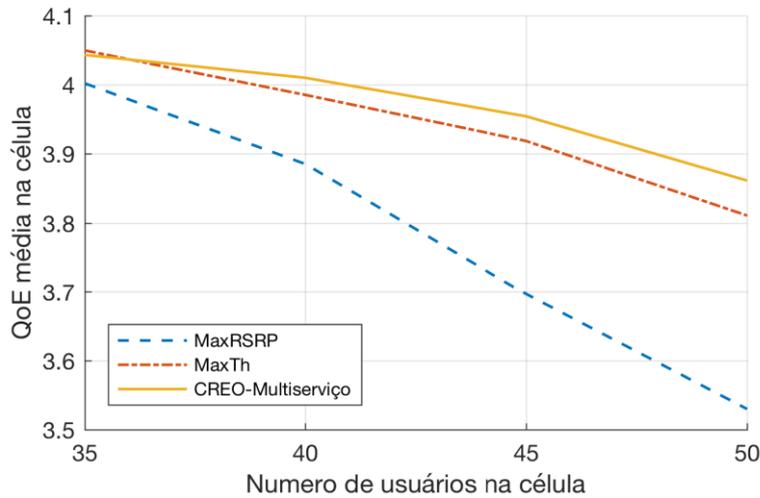


Figura 56. QoE média na célula por usuário Vs. Número de UEs ativos na célula

A seguir, apresentamos de forma detalhada o impacto da técnica proposta sobre cada tipo de serviço para o cenário com 45 UEs (Figuras abaixo), representando uma situação de forte congestionamento na MeNB.

A Figura 57, compara a QoE média de cada tipo de serviço. Nota-se neste caso de estudo, como a técnica CREO-Multi-serviço favoreceu em maior medida o serviço de vídeo (com ganhos na QoE de 10.41% vs. 7.61%) enquanto a técnica *MaxTh* beneficiou um pouco mais o serviço de dados (9.16% vs 8.24%). Observa-se também como o serviço de voz sofreu pequenas degradações (aproximadamente de 1%) nodesempenho médio pela expansão da cobertura da PeNB.

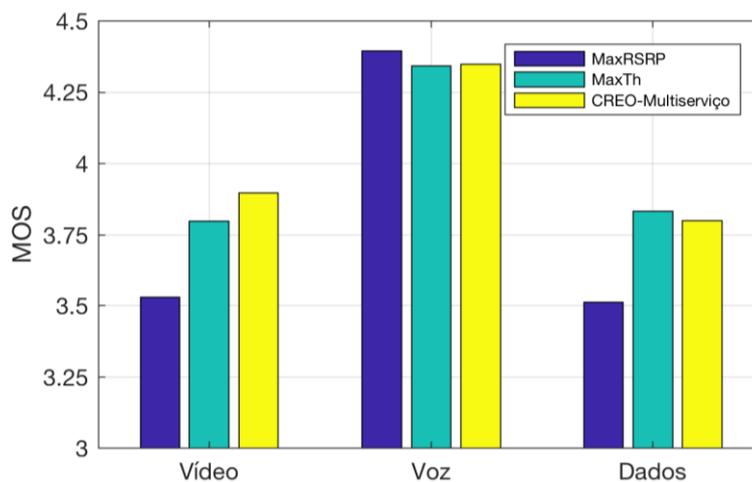


Figura 57. QoE média por tipo de serviço para o cenário com 45 UEs ativos na célula

As Figuras 58, 59 e 60 fornecem maiores informações estatísticas sobre o desempenho das sessões simuladas para todos os UEs da célula. A Figura 58 ilustra a função de densidade acumulada (FDA) para os valores de QoE obtidos pelos UEs de vídeo. Observa-se que a técnica *CREO-Multi-serviço* apresenta melhores valores para este tipo de serviço, mantendo o 66% dos UEs com uma MOS superior a 4.

A Figura 59, apresenta a FDA obtida para o desempenho dos UEs com uma sessão ativa de voz. Nota-se que pelo uso do CRE, alguns UEs foram prejudicados. Nestes experimentos, a percentagem de UEs que tiveram sua QoE com valores menores a 3 foi de 1.6% para a técnica *CRE-Multi-serviço* e do 2.2% para MaxTh.

No caso dos UEs com serviço de dados, a Figura 60 mostra como a abordagem *MaxTh* beneficiou em maior medida aos UEs com serviços de dados. Isto se deve a que este tipo de UEs apresentam uma maior taxa de transmissão (até 800 Mbps c/u), contribuindo com o aumento da vazão global da célula.

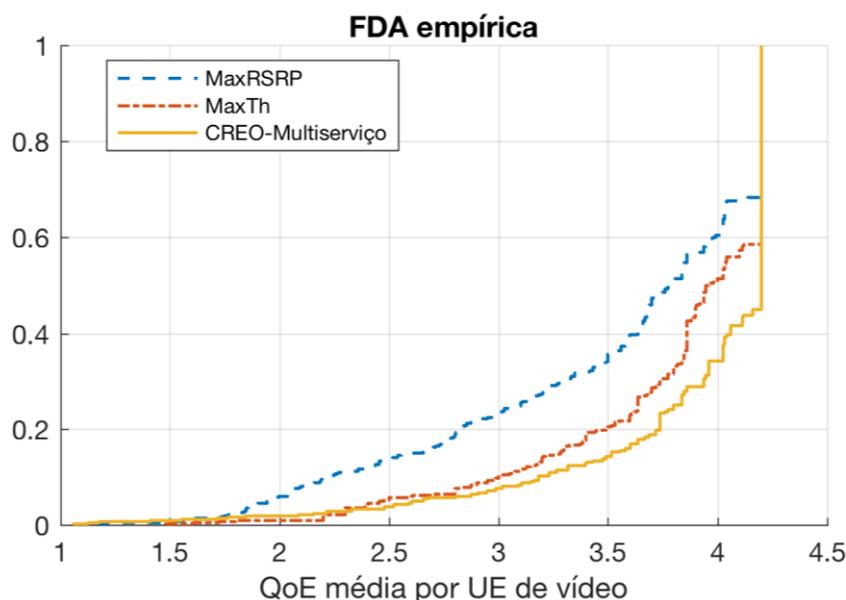


Figura 58. FDA empírica para a QoE média por usuário com serviço de vídeo (cenário com 45 UEs)

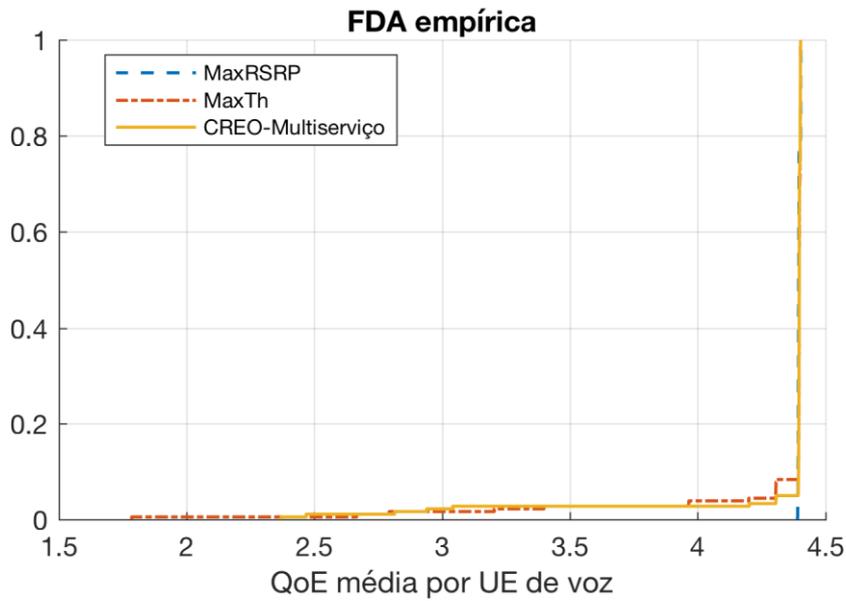


Figura 59. FDA empírica para a QoE média por usuário com serviço de voz (cenário com 45 UEs)

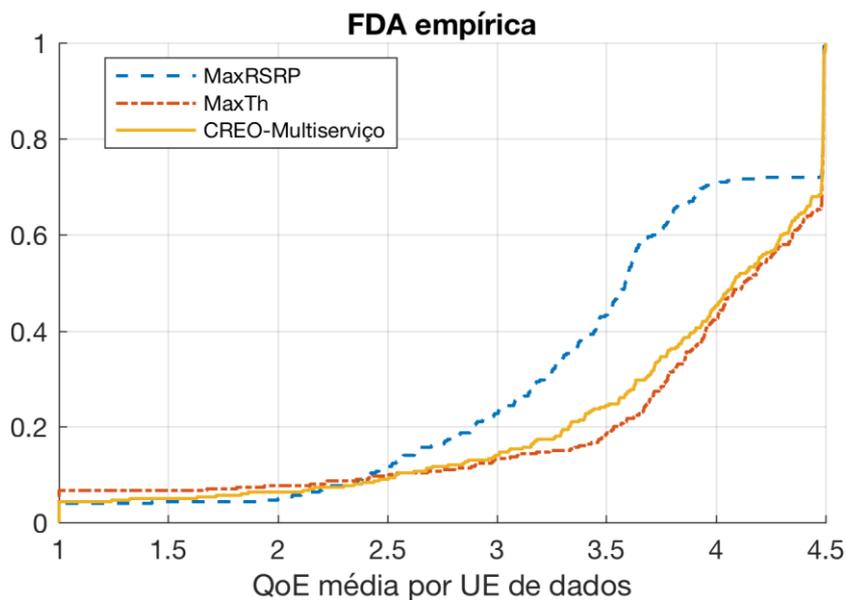


Figura 60. FDA empírica para a QoE média por usuário com serviço de dados (cenário com 45 UEs)

Da análise apresentada nesta seção, concluímos que a técnica proposta atingiu as metas de aprendizagem propostas, modificando os *CRE-offsets* para cada tipo de serviço conforme as condições de tráfego da célula a fim de melhorar a QoE dos UEs.

Por outro lado, destaca-se a utilização da estrutura hierárquica e a criação de subtarefas como um dos componentes chave para tratar o incremento do espaço de estados/ações no problema de aprendizagem.

No capítulo seguinte, apresenta-se as conclusões do trabalho assim como algumas sugestões para trabalhos futuros.

7

Conclusões e trabalhos futuros

7.1.

Conclusões

Nesta tese investigou-se como usar técnicas de aprendizagem para apoiar o desenvolvimento de mecanismos cognitivos/cooperativos de auto-organização em redes móveis heterogêneas. Com essa abordagem, procurou-se que a rede aprenda da experiência e realize de forma autônoma as adaptações necessárias para atingir os objetivos de desempenho desejados.

Para tal fim, inicialmente foram consolidados, a partir da literatura, os aspectos fundamentais e necessários para o desenvolvimento de mecanismos cognitivos bem como para a integração desses mecanismos em arquiteturas de acesso rádio móvel.

A partir desses conceitos fundamentais, foi desenvolvida a arquitetura distribuída denominada CCSON. CCSON foi concebida como uma arquitetura modular, expansível e modificável, cujos componentes são orientados às funcionalidades de auto-organização. Estas características facilitaram a implementação dos algoritmos das diferentes abordagens e técnicas de aprendizagem usadas neste trabalho.

Usando a arquitetura desenvolvida e tendo como foco o problema de balanceamento de carga entre macro e pico células, apresentou-se, no capítulo 5, duas técnicas de auto-organização com diferentes abordagens no seu processo de aprendizado (supervisionado e aprendizagem por reforço).

As técnicas desenvolvidas estiveram sustentadas na estratégia conhecida como *cell range expansion* (CRE), que permite aumentar a cobertura das PeNB, atraindo uma maior quantidade de UEs da macro para a pico-célula. Tais técnicas foram simuladas em um cenário com tráfego homogêneo, mostrando sua efetividade para melhorar o desempenho do sistema conforme o estado da célula.

Os resultados das simulações, mostraram que para o cenário simulado, foi possível aumentar o desempenho da célula aumentando o balanceamento de carga entre a macro e a pico-célula. Nesse cenário, em uma situação de congestionamento (40 usuários) obteve-se uma diminuição da perda de pacotes de 7.22% (MaxRSRP) para 4.67% e 4.15% para as técnicas com abordagem supervisionada e por reforço, respectivamente.

Embora os resultados encontrados tenham sido favoráveis nas duas técnicas, identificamos vantagens importantes na abordagem de aprendizagem por reforço. Enquanto o algoritmo supervisionado explorou a relação entre dados de entrada e saída de um conjunto de exemplos armazenados previamente, a abordagem por reforço realizou a aprendizagem a partir da interação com o ambiente sem precisar de um conjunto de dados de treinamento para operar.

A partir das observações anteriores, no capítulo 6 foi explorado o problema de balanceamento de carga cognitivo em redes móveis heterogêneas para um cenário que considera tráfego multi-serviço (vídeo, voz e descarga de arquivos) e medidas de qualidade centradas no usuário (QoE em termos da MOS). Naquele capítulo, foi proposta uma modificação na técnica CRE, agrupando os usuários pela classe de serviço utilizada e configurando valores de *offset* (usados para incrementar a cobertura da PeNB) diferentes para cada grupo de usuários conforme o estado da célula. O principal desafio enfrentado nesse problema multi-serviço foi o incremento do número de variáveis utilizadas para descrever o estado do sistema, causando por sua vez um incremento na complexidade na tarefa de aprendizagem. Para lidar com esse desafio, foi empregada uma metodologia fundamentada no princípio de "dividir para conquistar", descrevendo a solução do problema através de uma estrutura hierárquica de subtarefas de menor complexidade.

Os resultados das simulações mostraram a eficácia da técnica proposta para ajustar os *offsets*, seguindo uma política de otimização sequencial e com prioridade nos serviços de vídeo. Observou-se também como esta técnica centrada na avaliação da qualidade dos UEs permitiu abordar o problema multi-serviço de forma efetiva, considerando as características próprias de cada tipo de serviço e estabelecendo prioridades entre eles.

Com base no apresentado nesta tese, destaca-se a relevância na integração de mecanismos de aprendizagem nas funcionalidades de auto-organização e

gerência de recursos das futuras redes móveis. Esta abordagem permite um incremento da automatização e independência de operadores humanos para enfrentar o aumento na complexidade nas tarefas de gerenciamento e otimização da rede de acesso rádio-móvel.

Considera-se que a arquitetura e o desenvolvimento de software propostos nesta tese podem ser adotados por novos estudos, facilitando o desenvolvimento de técnicas de auto-organização sustentadas no uso de mecanismos de aprendizagem.

7.2. Trabalhos futuros

A partir dos desenvolvimentos e resultados nesta tese, é proposto como trabalho futuro a exploração no uso de técnicas emergentes na literatura em aprendizagem de máquina tais como *Deep Learning* [99] e suas aplicações envolvendo estratégias de aprendizagem por reforço. Estas técnicas podem oferecer novas formas de tratar o problema de balanceamento de carga multi-serviço empregando efetivamente um número maior de variáveis de estado.

Por outro lado, nos cenários e as técnicas desenvolvidas nesta tese, implementou-se a formação de grupos cognitivos formados por um agente que integrou as capacidades de aprendizagem e um outro agente cooperativo que forneceu informações complementares de desempenho na célula. Como trabalho futuro propomos analisar os processos necessários para os casos que precisem da formação automática de grupos com múltiplos agentes cooperativos e cognitivos. Nesse sentido, deve-se ter em conta a definição de funções de descobrimento, associação e interação com novos elementos de acesso, assim como também a adaptação dos mecanismos de aprendizagem e os modelos previamente aprendidos. Adicionalmente, propõe-se definir mecanismos para determinar em que situações é possível transferir as políticas aprendidas entre os elementos de acesso. Nesse caso, deve-se ter em conta que se as condições de operação são muito diferentes e será necessário realizar novos processos de aprendizado.

- [1] BHUSHAN, N. et al. Network densification: The dominant theme for wireless evolution into 5G, **IEEE Communications Magazine**, vol. 52, no. 2, pp. 82–89, 2014.
- [2] ALIU, O. G. et al. A Survey of Self Organisation in Future Cellular Networks, **IEEE Communications Surveys & Tutorials**, vol. 15, no. 1, pp. 336–361, 2013.
- [3] IMRAN, A.;ZOHA, A. Challenges in 5G: how to empower SON with big data for enabling 5G, **IEEE Network**, vol. 28, no. 6, pp. 27–33, Nov. 2014.
- [4] BOJOVIĆ, B. et al. Machine learning-based dynamic frequency and bandwidth allocation in self-organized LTE dense small cell deployments, **Eurasip Journal on Wireless Communications and Networking**, vol. 2016, no. 1, 2016.
- [5] HÄMÄLÄINEN, S. et al. **LTE Self-Organising Networks (SON)**. Chichester, UK: John Wiley & Sons, Ltd, 2011.
- [6] MITOLA, J.;MAGUIRE, G. Q. Cognitive radio: making software radios more personal, **IEEE Personal Communications**, vol. 6, no. 4, pp. 13–18, 1999.
- [7] MOYSEN, J.;GIUPPONI, L. **From 4G to 5G: Self-organized Network Management meets Machine Learning**, pp. 1–25, 2017.
- [8] KLAINÉ, P. V. et al. A Survey of Machine Learning Techniques Applied to Self Organizing Cellular Networks, **IEEE Communications Surveys & Tutorials**, vol. 19, no. 4, pp. 1–1, 2017.
- [9] THOMAS, R. W. et al. Cognitive networks, in **First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks**, 2005. DySPAN 2005., 2005, pp. 352–360.
- [10] KEPHART, J. O.;CHESS, D. M. The vision of autonomic computing, **Computer**, vol. 36, no. 1, pp. 41–50, Jan. 2003.

- [11] PETROVA, M. Cognitive wireless networks: your network just became a teenager, **25th IEEE International**, 2006.
- [12] LANGLEY, P. et al. Cognitive architectures: Research issues and challenges, **Cognitive Systems Research**, vol. 10, no. 2, pp. 141–160, Jun. 2009.
- [13] BALDO, N.;ZORZI, M. Cognitive network access using fuzzy decision making, **Wireless Communications, IEEE Transactions on**, vol. 8, no. 7, pp. 3523–3535, 2009.
- [14] AL-SAAD, A. et al. Cognitive network framework for heterogeneous wireless networks, **Procedia Computer Science**, vol. 60, no. 1, pp. 216–225, 2015.
- [15] AL-SAAD, A. et al. Semantic Reasoning in Cognitive Networks for Heterogeneous Wireless Mesh Systems, **IEEE Transactions on Cognitive Communications and Networking**, vol. 3, no. 3, pp. 374–389, Sep. 2017.
- [16] ZORZI, M. et al. Cognition-Based Networks: A New Perspective on Network Optimization Using Learning and Distributed Intelligence, **IEEE Access**, vol. 3, pp. 1512–1530, 2015.
- [17] ZORZI, M. et al. COBANETS: A new paradigm for cognitive communications systems, in **2016 International Conference on Computing, Networking and Communications (ICNC)**, 2016, no. Icnc, pp. 1–7.
- [18] WANG, X. et al. Artificial Intelligence-Based Techniques for Emerging Heterogeneous Network: State of the Arts, Opportunities, and Challenges, **IEEE Access**, vol. 3, pp. 1379–1391, 2015.
- [19] SHARMA, A. et al. Load balancing in Cellular Network: A review, in **Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference**, 2012, pp. 1–5.
- [20] LOBINGER, A. et al. Load balancing in downlink LTE self-optimizing networks, **IEEE Vehicular Technology Conference**, 2010.
- [21] RODRÍGUEZ, J. et al. Load balancing in a realistic urban scenario for LTE networks, **IEEE Vehicular Technology Conference**, 2011.
- [22] A.ATAYERO, A. et al. Neural-encoded Fuzzy Models for Load Balancing in 3GPP LTE, **International Journal of Applied**

- Information Systems**, vol. 4, no. 1, pp. 34–40, Sep. 2012.
- [23] MWANJE, S. S.;MITSCHLE-THIEL, A. A Q-learning strategy for LTE mobility Load Balancing, **IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC**, no. 5, pp. 2154–2158, 2013.
- [24] KUDO, T.;OHTSUKI, T. Cell range expansion using distributed Q-learning in heterogeneous networks, **EURASIP Journal on Wireless Communications and Networking**, vol. 2013, no. 1, p. 61, 2013.
- [25] L. DU, J. BIGHAM, L. CUTHBERT, C. PARINI, AND P. N. Using dynamic sector antenna tilting control for load balancing in cellular mobile communications,”, in **In International Conference on Telecommunications, ICT2002, Beijing, Citeseer**, pp. 344– 348, 2002.
- [26] GUO, W. et al. Spectral-and energy-efficient antenna tilting in a HetNet using reinforcement learning, **IEEE Wireless Communications and Networking Conference, WCNC**, pp. 767–772, 2013.
- [27] TOMFORDE, S. et al. Load-aware Reconfiguration of LTE-Antennas - Dynamic Cell-phone Network Adaptation Using Organic Network Control, in **Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics**, 2014, pp. 236–243.
- [28] ANDREWS, J. G. et al. An overview of load balancing in HetNets: Old myths and open problems, **IEEE Wireless Communications**, vol. 21, no. April, pp. 18–25, 2014.
- [29] RILEY, G. F.;HENDERSON, T. R. The ns-3 Network Simulator, in **Modeling and Tools for Network Simulation**, Springer, Ed. 2010, pp. 15–34.
- [30] BALDO, N. et al. An open source product-oriented LTE network simulator based on ns-3, **Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '11**, p. 293, 2011.
- [31] FRANCO, C. A. S. et al. A cognitive and cooperative SON framework for 5G mobile radio access networks, in **2016 IEEE Globecom**

- Workshops**, GC Wkshps 2016 - Proceedings, 2016.
- [32] SIERRA, C. A.;MARCA, R. B. DE. Balanceamento de carga cognitivo em redes móveis heterogêneas e auto-organizáveis, in **SBrT 2016**, 2016.
- [33] FRANCO, C. A. S.;DE MARCA, J. R. B. Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach, in 2015 7th **IEEE Latin-American Conference on Communications (LATINCOM)**, 2015, pp. 1–5.
- [34] AGOULMINE, N. **Autonomic network management principles: from concepts to applications**. Elsevier, 2011.
- [35] SPILLING, A. G. et al. Self-organisation in future mobile communications, **Electronics & Communication Engineering Journal**, vol. 12, no. 3, pp. 133–147, 2000.
- [36] YANMAZ, E. et al. Self-Organization in Cellular Wireless Networks via Fixed Relay Nodes, in **Global Telecommunications Conference**, 2006. GLOBECOM '06. IEEE, 2006, pp. 1–5.
- [37] PREHOFER, C.;BETTSTETTER, C. Self-organization in communication networks: principles and design paradigms, **Communications Magazine, IEEE**, vol. 43, no. 7, pp. 78–85, 2005.
- [38] ELMENREICH, W.;DE MEER, H. **Self-organizing networked systems for technical applications: A discussion on open issues**. Springer, 2008.
- [39] NOKIA. **Intelligent Self-Organizing Networks**: Increased automation for higher network performance with lower costs (With Paper), 2013.
- [40] MILLS, K. L. A brief survey of self- organization in wireless sensor networks, **Wireless Communications and Mobile Computing**, vol. 7, no. 7, pp. 823–834, 2007.
- [41] DRESSLER, F. A study of self-organization mechanisms in ad hoc and sensor networks, **Computer Communications**, vol. 31, no. 13, pp. 3018–3029, 2008.
- [42] ALLIANCE, N. NGMN recommendation on SON and O&M requirements, **Next Generation Mobile Networks, White paper, December**, 2008.

- [43] 3GPP. **TR 32.902: Self-configuring and self-optimising network use cases and solutions** (release 9). 2010.
- [44] 3GPP. **TS 32.501: Self-configuration of network elements; Concepts and Requirements**, (Release 9). 2010.
- [45] EISENBLATTER, A. et al. Self-configuration in LTE radio networks: automatic generation of eNodeB parameters, in **Vehicular Technology Conference (VTC Spring)**, 2011 IEEE 73rd, 2011, pp. 1–3.
- [46] SANNECK, H. et al. Context based configuration management of plug and play LTE base stations, in **Network Operations and Management Symposium (NOMS)**, 2010 IEEE, 2010, pp. 946–949.
- [47] NAWROCKI, M. J. et al. **Understanding UMTS radio network modelling, planning and automated optimisation**. Wiley Online Library, 2008.
- [48] 3GPP. **TS 32.541: Telecommunication management; Self-Organizing Networks (SON)**; Self-healing concepts and requirements, 2010.
- [49] POOLE, D. et al. **Computational intelligence: a logical approach**. Oxford University Press, 1998.
- [50] RUSSELL, S.;NORVIG, P. **Artificial Intelligence: A Modern Approach**. Pearson, 2014.
- [51] MACKWORTH, A. K. Constraint satisfaction problems, **Encyclopedia of AI**, vol. 285, p. 293, 1992.
- [52] PUTERMAN, M. L. **Markov decision processes: discrete stochastic dynamic programming**. John Wiley & Sons, 2014.
- [53] KOTSIANTIS, S. B. et al. Supervised machine learning: A review of classification techniques, **Informatica**, vol. 31, pp. 249–268, 2007.
- [54] HAYKIN, S. S. et al. **Neural networks and learning machines**, vol. 3. Pearson Education Upper Saddle River, 2009.
- [55] CRISTIANINI, N.;SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. Cambridge university press, 2000.
- [56] NIELSEN, T. D.;JENSEN, F. V. **Bayesian networks and decision graphs**. Springer Science & Business Media, 2009.

- [57] BISHOP, C. M. **Pattern recognition and machine learning**, vol. 4. springer New York, 2006.
- [58] PRIEBE, C. E.; MARCHETTE, D. J. Adaptive mixture density estimation, **Pattern Recognition**, vol. 26, no. 5, pp. 771–785, 1993.
- [59] KASKI, S. Data exploration using self-organizing maps, in **Acta polytechnica scandinavica: mathematics, computing and management in engineering series no. 82**, 1997.
- [60] GOLUB, G. Numerical methods for solving linear least squares problems, **Numerische Mathematik**, vol. 7, no. 3, pp. 206–216, 1965.
- [61] Kaelbling, L. P. et al. Reinforcement Learning : A Survey, **Journal of Artificial Intelligence Research**, vol. 4, pp. 237–285, 1996.
- [62] BELLMAN, R. **Dynamic Programming**, Princeton, NJ: Princeton Univ, 1957.
- [63] HAYKIN, S. Cognitive radio: Brain-empowered wireless communications, **IEEE Journal on Selected Areas in Communications**, vol. 23, no. 2, pp. 201–220, 2005.
- [64] THATHACHAR, M. A L.; SASTRY, P. S. **Networks of Learning Automata**. Boston, MA: Springer US, 2004.
- [65] HU, Y. et al. Mobile Edge Computing—A Key Technology Towards 5G, **ETSI White Paper**, 2015.
- [66] HURSCH, W. L.; LOPES, C. V. Separation of Concerns, **Computer**, pp. 1–20, 1995.
- [67] BANDH, T. et al. Policy-based coordination and management of SON functions, **Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, IM 2011**, pp. 827–840, 2011.
- [68] ROBINSON, S. **Simulation: the practice of model development and use**, 2nd ed. Palgrave, 2014.
- [69] BALDO, N. et al. A new model for the simulation of the LTE-EPC data plane, in **Proceedings of workshop on ns-3 (WNS3)**, in conjunction with ICST SIMUTOOLS, 2012.
- [70] FRANCO, C. A. S.; DE MARCA, J. R. B. An open-source simulation of QoE for video applications in relay enhanced LTE-A networks, in

- 2014 11th **International Symposium on Wireless Communications Systems (ISWCS)**, 2014, pp. 406–410.
- [71] 3GPP. **TS 36.423: Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)**, 2014.
- [72] 3GPP **TR 36.814. Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (Release 9)**, 2010.
- [73] MONTGOMERY, D. C. et al. **Introduction to linear regression analysis**, vol. 821. John Wiley & Sons, 2012.
- [74] 3GPP. **TR 26.944: End-to-end multimedia services performance metrics**, 2017.
- [75] KLEINBAUM, D. et al. **Applied regression analysis and other multivariable methods**. Cengage Learning, 2013.
- [76] GLORENNEC, P. Y.; JOUFFE, L. Fuzzy Q-Learning, **Artificial Intelligence**, no. 3, pp. 486–491, 1997.
- [77] TIAN, P. et al. An adaptive bias configuration strategy for range extension in LTE-advanced heterogeneous networks, **IET International Conference on Communication Technology and Application (ICCTA 2011)**, pp. 336–340, 2011.
- [78] WU, Q. et al. Traffic-Aware Online Network Selection in Heterogeneous Wireless Networks, **IEEE Transactions on Vehicular Technology**, vol. 65, no. 1, pp. 381–397, Jan. 2016.
- [79] AKHTAR, Z.; FALK, T. H. Audio-Visual Multimedia Quality Assessment: A Comprehensive Survey, **IEEE Access**, vol. 5, pp. 21090–21117, 2017.
- [80] ZHAO, T. et al. QoE in Video Transmission: A User Experience-Driven Strategy, **IEEE Communications Surveys and Tutorials**, vol. 19, no. 1, pp. 285–302, 2017.
- [81] ITU-T. Amendment 1: Recommendation P.10/G.100. New Appendix I – Definition of Quality of Experience (QoE), **Telecommunication Standardization Sector of Itu-T**, vol. 100, no. 2006, 2007.
- [82] ALRESHOODI, M.; WOODS, J. Survey on QoE\QoS Correlation Models for Multimedia Services, **International Journal of Distributed and Parallel systems**, vol. 4, no. 3, pp. 53–72, 2013.

- [83] SHAHID, M. et al. No-reference image and video quality assessment: a classification and review of recent approaches, **EURASIP Journal on Image and Video Processing**, vol. 2014, no. 1, pp. 1–32, 2014.
- [84] KHAN, A. et al. Quality of experience-driven adaptation scheme for video applications over wireless networks, **IET Communications**, vol. 4, no. 11, p. 1337, 2010.
- [85] ITU-T REC. G.107. **The E-model: a computational model for use in transmission planning**, 2015.
- [86] KELLY, F. Charging and rate control for elastic traffic, **European Transactions on Telecommunications**, vol. 8, no. 1, pp. 33–37, Jan. 1997.
- [87] 3GPP **TR 32.425. Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)**, 2016.
- [88] BARTO, A. G.; MAHADEVAN, S. Recent Advances in Hierarchical Reinforcement Learning, **Discrete Event Dynamic Systems**, vol. 13, pp. 341–379, 2003.
- [89] HENRIQUE, C.; RIBEIRO, C. **Aprendizado por Reforço**, pp. 28–72, 1999.
- [90] SUTTON, R. S.; BARTO, A. G. Reinforcement Learning: An Introduction, **IEEE Transactions on Neural Networks**, vol. 9, no. 5, pp. 1054–1054, Sep. 1998.
- [91] CHUNMING LIU et al. Multiobjective Reinforcement Learning: A Comprehensive Overview, **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, vol. 45, no. 3, pp. 385–398, Mar. 2015.
- [92] TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control, **Systems, Man and Cybernetics, IEEE Transactions on**, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [93] CISCO. **Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021**, 2017.
- [94] KLAUE, J. et al. Evalvid: A framework for video transmission and quality evaluation, **Evaluation, Modelling Techniques and Tools**, no. September, 2003.
- [95] PINHEIRO, B. **Evalvid module by GERCOM**,

- <https://gitlab.com/gercom/evalvid-ns3>. 2013.
- [96] ARIZONA STATE UNIVERSITY. **YUV Video Sequences**, disponível em <http://trace.eas.asu.edu/yuv/index.html>. .
- [97] KHAN, S. et al. MOS-based multiuser multiapplication cross-layer optimization for mobile multimedia communication, **Advances in Multimedia**, vol. 2007, no. 1, 2007.
- [98] TANG, P. et al. QoE-Based Resource Allocation Algorithm for Multi-Applications in Downlink LTE Systems, **Proceedings of the 2014 International Conference on Computer, Communications and Information Technology**, no. Ccit, 2014.
- [99] SCHMIDHUBER, J. Deep Learning in neural networks: An overview, **Neural Networks**, vol. 61, pp. 85–117, 2015.