UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

CAMILA SOARES BARBOSA

Algoritmos Baseados em Estratégia Evolutiva para a Seleção Dinâmica de Espectro em Rádios Cognitivos

Goiânia
2013

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

AUTORIZAÇÃO PARA PUBLICAÇÃO DE DISSERTAÇÃO EM FORMATO ELETRÔNICO

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos "reproduzir" e "publicar" conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Algoritmos Baseados em Estratégia Evolutiva para a Seleção Dinâmica de Espectro em Rádios Cognitivos

Autor(a): Camila Soares Barbosa

Goiânia, 22 de Novembro de 2013.

Camila Soares Barbosa – Autor

Dr. Kleber Vieira Cardoso – Orientador

CAMILA SOARES BARBOSA

Algoritmos Baseados em Estratégia Evolutiva para a Seleção Dinâmica de Espectro em Rádios Cognitivos

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Computação.

Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Kleber Vieira Cardoso

Goiânia
2013

CAMILA SOARES BARBOSA

Algoritmos Baseados em Estratégia Evolutiva para a Seleção Dinâmica de Espectro em Rádios Cognitivos

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Computação, aprovada em 22 de Novembro de 2013, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Kleber Vieira Cardoso
Instituto de Informática – UFG
Presidente da Banca

Profa. Dr.ª Sand Luz Corrêa
Instituto de Informática – UFG

Prof. Dr. Celso Gonçalves Camilo Junior
Instituto de Informática – UFG

Prof. Dr. Aldri Luiz dos Santos
Departamento de Informática – UFPR

Camila Soares Barbosa

Graduada em Engenharia de Computação pela Universidade Federal de Goiás e em Redes de Comunicação pelo Centro Federal de Educação Tecnológica de Goiás, especialista em Segurança em Redes de Computadores pela Faculdade de Tecnologia SENAI de Desenvolvimento Gerencial. Atua como servidora pública federal na área de infraestrutura e segurança da informação.

À minha família, por prover as condições, ambiente e apoio necessários durante todo esse tempo.

Agradecimentos

A Deus pela saúde e disposição para prosseguir na conquista de objetivos.
Aos meus pais, Ruy e Catarina, por todo o amor e dedicação que me oferecem sempre.
Às minhas irmãs, Ursula e Fabiana, pela amizade e atenção.
Ao Prof. Kleber Cardoso, por sua maestria, sabedoria e paciência na orientação desse trabalho.
Aos Profs. Sand Luz, Rogério Salvini e Vinícius Borges pela parceria e pelas contribuições, proporcionando a conquista de importantes publicações.
Ao Prof. Celso, grande mestre que sabiamente despertou meu interesse para a área de Inteligência Artificial.
Ao mestre William Divino Ferreira, com carinho, pelo incentivo, crédito e apoio.
Aos Profs. Celso e Aldri pela presença na banca e contribuições à dissertação.
Aos colegas do grupo de pesquisa Labora: Micael, André Lauar, Diego, Pedro e todos os demais; pela amizade e apoio.
À equipe da secretaria: Edir, Mirian e Ricardo e todos os demais; pela atenção e suporte operacional.
Ao INF/UFG, pelas instalações e equipamentos utilizados.

“Todas as vitórias ocultam uma abdicação”

Simone de Beauvoir,
Memórias de uma jovem bem comportada - Mémoires dúne jeune fille rangée.

Resumo

Barbosa, Camila Soares. **Algoritmos Baseados em Estratégia Evolutiva para a Seleção Dinâmica de Espectro em Rádios Cognitivos**. Goiânia, 2013. 78p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Um dos principais desafios da Seleção Dinâmica de Espectro em Rádios Cognitivos é a escolha da faixa de frequência para cada transmissão. Essa escolha deve minimizar a interferência em dispositivos legados e maximizar a descoberta das oportunidades ou espaços em branco. Há várias soluções para essa questão, sendo que algoritmos de Aprendizado por Reforço são as mais bem sucedidas. Entre eles destaca-se o *Q-Learning*, cujo ponto fraco é a parametrização, uma vez que ajustes são necessários para que se alcance, com sucesso, o objetivo proposto. Nesse sentido, este trabalho propõe um algoritmo baseado em Estratégia Evolutiva e apresenta como características principais a adaptabilidade ao ambiente e a menor quantidade de parâmetros. Através de simulação, o desempenho do *Q-Learning* e da proposta deste trabalho foram comparados em diversos cenários. Os resultados obtidos permitiram avaliar a eficiência espectral e a adaptabilidade ao ambiente. A proposição deste trabalho apresentou resultados promissores na maioria dos cenários.

Palavras–chave

Rádios Cognitivos, Seleção Dinâmica de Espectro, Estratégias Evolutivas

Abstract

Barbosa, Camila Soares. **Algorithms Based on Evolutionary Strategy for Dynamic Spectrum Selection in Cognitive Radios**. Goiânia, 2013. 78p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

One of the main challenges in Dynamic Spectrum Selection for Cognitive Radios is the choice of the frequency range for each transmission. This choice should minimize interference with legacy devices and maximize the discovering opportunities or white spaces. There are several solutions to this issue, and Reinforcement Learning algorithms are the most successful. Among them stands out the Q-Learning whose weak point is the parameterization, since adjustments are needed in order to reach successfully the proposed objective. In that sense, this work proposes an algorithm based on evolutionary strategy and presents the main characteristics adaptability to the environment and fewer parameters. Through simulation, the performance of the Q-Learning and the proposal of this work were compared in different scenarios. The results allowed to evaluate the spectral efficiency and the adaptability to the environment. The proposal of this work shows promising results in most scenarios.

Keywords

Cognitive Radio, Dynamic Spectrum Selection, Evolutionary Strategies

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

CAPÍTULO 1

Introdução

Com a evolução tecnológica, é latente a necessidade por mais faixas de frequência, porém esse crescimento pode esbarrar na escassez de recursos e nas limitações provocadas pela alocação estática do espectro. Diante disso, esse modelo é reconhecidamente incapaz de atender às demandas das tecnologias de rede sem fio emergentes [4], pois impede que espaços em branco no espectro sejam aproveitados.

Para melhor aproveitamento e racionalização de uso dos recursos, faz-se necessário o emprego de uma nova abordagem de utilização do espectro que não limite a evolução e o surgimento de novas tecnologias de transmissão sem fio. Nesse sentido, a Seleção Dinâmica de Espectro (DSS – *Dynamic Spectrum Selection*) foi criada para ser um modelo complementar à alocação estática com finalidade de oferecer maior eficiência na utilização do espectro.

1.1 Motivação

O atual modelo de alocação de espectro é fundamentado na reserva estática das faixas de frequência na forma de concessões entre o Estado, detentor do recurso, e o interessado na exploração de modo exclusivo. Diante desse contexto, o contratante da faixa de espectro possui direitos e por isso, uma transmissão legítima não pode ser prejudicada por transmissões não autorizadas nessa mesma faixa. Assim, é necessário a existência de regras de regulação e proteção do espectro, de modo a prevalecer o contrato firmado entre o Estado e aquele que tem direito de explorar o recurso.

O espectro de frequência tem se tornado um recurso escasso devido ao modelo de alocação estática e à crescente demanda tecnológica. Embora o espectro seja amplo, frequências extremas implicam fenômenos inerentes às suas características e podem prejudicar a qualidade de transmissão. Por exemplo, em baixas frequências a taxa máxima de dados é pequena e, consequentemente, a capacidade de transmissão é limitada. Isso se deve à perda de potência do sinal em relação à distância da origem quando o meio de propagação é o ar. As altas frequências são mais suscetíveis a fenômenos como a difração e refração, devido à perda de energia pela absorção do sinal.

Assim, a porção do espectro que confere características mais adequadas para transmissão concentra-se nas faixas intermediárias e coincide com a porção regulamentada dentro do modelo de alocação estática e são, portanto, faixas bastante valorizadas. No Brasil, isso pode ser ilustrado pelo valor de $R\$2,71$ bilhões arrecadado em lances no leilão de faixas para tecnologia 4G, promovidas pela Agência Nacional de Telecomunicações (Anatel) no ano de 2012[1]. Nos Estados Unidos, no ano de 2008, em plena recessão americana, a Comissão Federal de Comunicação (FCC – *Federal Communications Commission*) arrecadou o valor de $US\$19$ bilhões pela faixa de 700 MHz[2].

Duas abordagens para oportunidade de transmissão devem ser levadas em consideração, o espacial e o temporal. A primeira é mais factível uma vez que apenas requer um levantamento de dados baseado em região, considera-se uma solução parcial o aproveitamento das oportunidades espaciais, ou seja, utilizar as bandas de frequências que não estão reservadas em uma determinada região com o auxílio de bases de dados e hardware para georreferenciamento [46]. A segunda abordagem, embora de maior complexidade de implementação, é onde reside maior quantidade de espectro disponível, pois a média efetiva em termos de ocupação das faixas de frequências é inferior a 20% [25, 43, 53].

Nesse cenário, é possível constatar a existência de espaços em branco no espectro, que são instantes em que os proprietários da faixa de espectro não a estão utilizando e poderia ser ofertada a outros tipos de dispositivos que a queiram utilizar para transmitir dados. Essa abordagem também tem como premissa a possibilidade de aproveitamento das oportunidades de transmissão, desde que respeitadas as condições de uso. Sua aplicação fica limitada a cenários onde a maior parte ou todo o espectro útil para transmissões eletromagnéticas se encontra licenciado, como ocorre em grandes centros urbanos [25, 53]. Nesse tipo de cenário, a alocação espacial pode indicar pouca ou nenhuma disponibilidade de espaços em branco.

O percentual de ocupação das faixas de espectro nunca é total pois não se transmite o tempo todo de todos os lugares, reforçando a hipótese da existência de oportunidades temporais de transmissão. Porém, não há garantias de que os buracos existentes no espectro tenham características mínimas que permitam sua utilização. Além disso, há evidências de que, embora exista a previsão de uso para uma determinada finalidade, essa faixa nunca seja utilizada ou esteja subutilizada. Porém, isso não implica na perda do direito de uso pelo proprietário daquela faixa de frequência. Existem, portanto, muitas oportunidades para transmissão, mas é necessário que os dispositivos estejam preparados para aproveitá-las dinamicamente ao longo do tempo.

No contexto apresentado, para que a seleção dinâmica ao espectro seja efetiva, é necessário que os dispositivos sejam capazes de alterar dinamicamente sua faixa de

[1] http://agenciabrasil.ebc.com.br/noticia/2012-06-13/anatel-arrecada-r-29-bilhoes-com-leilao-de-4g
[2] http://wireless.fcc.gov/auctions/default.html?job=auction_summary&id=73

frequência em busca de oportunidades de transmissão. Idealmente, cada dispositivo deveria ser um rádio cognitivo (CR – *Cognitive Radio*), conforme proposto em [45, 29], sendo dotado de um ciclo de cognição que interaja com o ambiente de operação para observar, aprender, planejar, decidir e atuar. Apesar dos dispositivos atuais para acesso dinâmico ao espectro ainda estarem distantes do seu modelo ideal, eles já são capazes de utilizar o espectro disponível de maneira mais eficiente.

1.2 Estado da arte da DSS

Conforme a nomenclatura sugerida em [4], a DSS é uma das ações realizadas por um Usuário Secundário (SU – *Secondary User*) com o intuito de aproveitar as oportunidades de transmissão deixadas pelos Usuário Primário (PUs – *Primary User*) e, portanto, é um problema relacionado à função de Gerenciamento de Espectro.

Basicamente, a DSS tem como objetivo principal a escolha de uma faixa de frequência que atenda às necessidades das aplicações do SU, mantendo a interferência sobre os PUs dentro do limite tolerável. Sob a perspectiva de um SU, cada faixa de frequência ou canal pode ter uma largura de banda diferente, uma vez que os CRs são capazes de reconfigurar vários de seus parâmetros de comunicação. De fato, a DSS é apenas um dos aspectos que envolvem redes de CRs e que tem recebido atenção significativa. Há vários trabalhos sobre esse tema, empregando diferentes abordagens como teoria dos jogos [56], modelos de previsão [21], soluções bio-inspiradas [12, 15] e aprendizado por reforço [20, 62]. O aprendizado por reforço tem se destacado como uma abordagem promissora, alcançando resultados satisfatórios inclusive em testes com dispositivos reais [48].

Em cada canal, pode haver um nível diferente de ocupação de acordo com o PU licenciado para uso da banda. Características de propagação do sinal como multipercurso, desvanecimento, fontes de interferência/ruído, dentre outras também levam a diferenças entre os canais sob a perspectiva do SU. Assim, as condições para transmissão do SU em cada canal podem variar de maneira significativa em diferentes escalas de tempo. Em geral, os trabalhos sobre DSS visam atender a demanda por banda do SU, buscando fazer a melhor escolha [51, 57, 66]. Agindo assim, o dispositivo prioriza a eficiência na transmissão alcançando maior qualidade de serviço [28].

Os autores de [66] estudam o problema de acesso oportunístico do SU ao espectro sobre múltiplos canais através do Processo de Decisão de Markov parcialmente observável. Mais recentemente, [14] apresentou uma solução ideal para escalonar pacotes SU em um sistema multi-canal baseado em uma análise de tempo de espera.

Sistemas de DSS que envolvem múltiplos SUs competindo podem ser classificados em: com coordenação e sem coordenação. Na abordagem coordenada, múltiplos

SUs são coordenados para acessar o espectro por meio de um canal de controle comum. Em [65], os SUs se organizam em grupos, com base na similaridade entre os canais disponíveis usando canais comuns disponíveis localmente. Em [51], a coordenação entre CRs é formulado como um problema que envolve uma relação conjunta potência/taxa de controle e otimização de atribuição de canais. Em [11], cada SU recomenda aos demais SUs aqueles canais em que alcançou sucesso, caracterizando a cooperação. Existem alguns desafios nas pesquisas de sistemas de DSS coordenados que empregam modelos de comercialização do espectro. Em [16] é feito um levantamento sobre preços, acordos e contratos de leilão.

As principais desvantagens de sistemas de acesso ao espectro com coordenação para os sem coordenação são respectivamente: a necessidade de uma entidade central e a sobrecarga de tráfego. Além disso, não é raro encontrar ambientes de produção, onde os donos de redes diferentes (por exemplo, redes 802.11) não estão dispostos a cooperar entre si.

Na abordagem sem coordenação, múltiplos SUs competem de uma forma egoísta pelos buracos em branco. Nesse contexto, a teoria dos jogos é um arcabouço amplamente empregado e alguns estudos o têm utilizado para modelar as interações entre o SUs [18]. O estado da arte em soluções para os sistemas de DSS sem coordenação emprega algoritmos de aprendizado por reforço. Em [63], o algoritmo *Q-Learning* é utilizado para estabelecer um esquema de DSS. Posteriormente, esse mecanismo foi implementado na plataforma *GNU Radio* e alcançou resultados promissores [48].

O aprendizado por reforço é um conjunto de técnicas do tipo *online* que não depende de nenhum conhecimento prévio do ambiente em que irá operar, sendo essa uma das motivações para sua aplicação na DSS. Outra vantagem importante das técnicas de aprendizado por reforço, é que a ação ótima é encontrada desde que o mecanismo execute durante um número suficiente de interações com o ambiente. O *Q-Learning* é um algoritmo representante dessa classe que apresenta melhores resultados na solução desse problema. No entanto, em geral, as técnicas baseadas em aprendizado por reforço apresentam uma quantidade de parâmetros cujo ajuste inadequado pode afetar severamente seu desempenho. Por exemplo, a escolha de uma alta taxa de exploração pode ser adequada para um cenário muito dinâmico, mas inapropriada para um cenário com menor dinamicidade. As faixas ou canais do espectro podem apresentar diferentes condições em diferentes escalas de tempo. Nesse contexto, é necessário escolher entre ajustes regulares dos parâmetros, que é geralmente uma tarefa não trivial, ou tolerar períodos de uso ineficiente das oportunidades de transmissão.

Por isso, para resolução do problema de DSS, esse trabalho mostra um novo algoritmo inspirado em estratégia evolutiva. Este algoritmo visa maximizar o aproveitamento das oportunidades de transmissão do espectro em ambiente dinâmico e minimizar a

necessidade de ajuste de parâmetros independente do cenário em que o dispositivo opere. Dessa forma, o algoritmo projetado tem a característica de ser adaptável a diversos ambientes e sem necessidade de configuração, surgindo como uma solução competitiva.

O mecanismo descrito neste trabalho promove o aproveitamento da oportunidade identificada, permitindo a utilização do espectro de modo otimizado, pela oferta de decisões mais acertadas ao rádio. Assim, a proposta provê um mecanismo cognitivo para que os dispositivos possam coexistir com os usuários licenciados de modo a minimizar o impacto do uso do espectro compartilhado e maximizar o aproveitamento do espectro disponível. Além de oferecer uma decisão mais adequada, proporciona ainda característica de reconfigurabilidade pelo constante monitoramento e reação rápida diante de alterações no contexto em que o dispositivo encontra-se inserido.

1.3 Objetivos

Este trabalho apresenta como proposta uma abordagem evolutiva para a seleção dinâmica de espectro. Nesse sentido, foram estabelecidos os seguintes objetivos:

1. Estudar o problema de seleção dinâmica de espectro;
2. Estudar o *Q-Learning*, algoritmo mais bem sucedido aplicado ao problema investigado, de modo que suas características fossem detalhadas e o potencial de melhoria fosse avaliado;
3. Buscar técnicas de computação evolucionária que apresentassem potencial para solução do problema;
4. Elaborar algoritmos evolucionários para o aproveitamento dinâmico do espectro em diferentes cenários.

1.4 Organização da dissertação

Esta dissertação está organizada da seguinte forma:

- No capítulo 2, são apresentados alguns conceitos sobre rádios cognitivos com destaque para o ciclo cognitivo e reconfigurabilidade;
- No capítulo 3, são descritas a teoria a respeito de Aprendizado por Reforço e Estratégia Evolutiva;
- No capítulo 4, são apresentadas a representação do problema, as estratégias de implementação adotadas neste trabalho, as etapas de evolução do novo algoritmo proposto e as características agregadas a cada evolução;
- No capítulo 5, são descritos os cenários e a plataforma de simulação, são apresentados o algoritmo *Q-Learning* utilizado nesse trabalho, a avaliação do impacto de

seus parâmetros e a análise comparativa sobre diversos aspectos entre os mecanismos avaliados;

- Por fim, o capítulo 6 apresenta a conclusão e as perspectivas de trabalhos futuros.

CAPÍTULO 2

Rádios cognitivos: capacidade cognitiva e reconfigurabilidade

Rádios Cognitivos são a principal tecnologia que permite Redes da Próxima Geração - NGN (*Next Generation Networks*) utilizar o espectro de forma dinâmica. Trata-se de um novo paradigma de comunicação para aproveitamento do espectro de modo oportunístico através da alteração de seus parâmetros de transmissão/recepção de acordo com o ambiente em que está em operação. Este capítulo descreve essa tecnologia, destacando algumas de suas características essenciais: capacidade cognitiva e reconfigurabilidade. Esses conceitos são importantes para a compreensão dos assuntos abordados neste trabalho e também para ilustrar a necessidade de novas soluções para o problema de Seleção Dinâmica de Espectro.

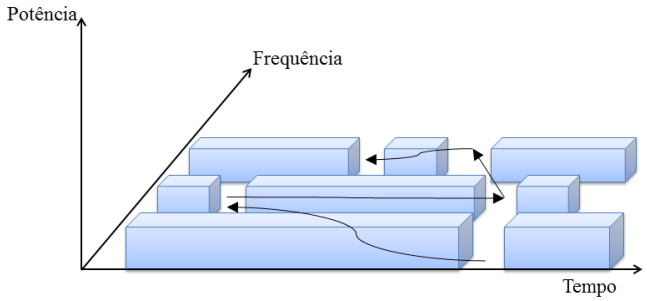


Figura 2.1: *Representação conceitual dos espaços em branco no espectro.*

2.1 Rádios cognitivos

Rádio Cognitivo, como o próprio nome sugere, é um dispositivo de comunicação dotado de algum tipo de inteligência para lidar com as variações e adversidades do meio em que está inserido, através do ajuste de parâmetros de transmissão.

O aproveitamento oportunístico está relacionado à capacidade de utilizar as oportunidades de transmissão representadas pelos espaços em branco no espectro através da

seleção dinâmica de espectro. Para alcançar essa capacidade, é necessário conferir flexibilidade de configuração ao dispositivo. Contudo, tecnologias que aproveitem oportunidades de transmissão, tanto no espaço quanto no tempo, devem ser capazes de perceber quando o PU retorna a utilizar sua faixa e o SU deve possuir a habilidade de responder prontamente para a preservação das transmissões dos PUs. A Figura 2.1 ilustra os espaços em branco disponíveis em uma porção de espectro e as suas variações no tempo.

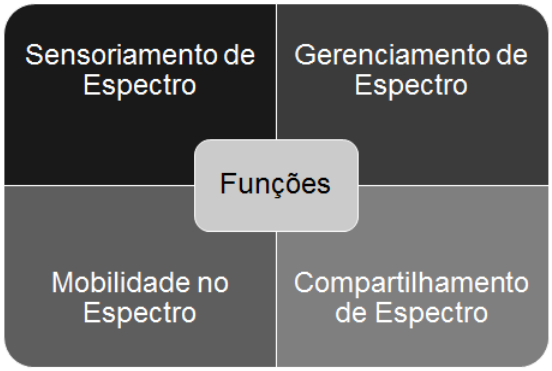


Figura 2.2: *Funções desempenhadas pelos RCs.*

Segundo a arquitetura apresentada em [4], as funções desempenhadas por um Rádio Cognitivo estão representadas na Figura 2.2 e podem ser resumidas em:

- Sensoriamento de Espectro - busca informações sobre as porções disponíveis no espectro de modo a prever a presença do PU e sem interferir na reconfiguração do dispositivo de rádio cognitivo ou nos dispositivos primários. Trata-se de uma etapa prévia para reconhecimento do ambiente. Existem mecanismos de sensorimamento de espectro que tratam da avaliação do espectro, acompanhando suas mudanças e apenas informando a situação atual a outro mecanismo que permitirá a transmissão.
- Gerenciamento de Espectro - atua analisando o espectro disponível para atender às necessidades de comunicação do usuário, utilizando-se das informações obtidas pela função de sensoriamento de espectro. Assim, é possível selecionar o melhor canal disponível para transmitir.
- Mobilidade no Espectro - promove a manutenção dos requisitos de comunicação mesmo diante da necessidade de transição para outra porção do espectro que esteja melhor. É uma função que capacita o dispositivo de rádio cognitivo a desocupar o canal quando o PU é detectado.
- Compartilhamento de Espectro - atua proporcionando o escalonamento justo no uso do espectro entre os usuários que queiram utilizá-lo. Esta é, portanto, uma função que desempenha o papel de coordenação entre os usuários que disputam os canais.

A partir das ideias de Mitola, foram definidas duas características principais para o rádio cognitivo [28]: a Capacidade Cognitiva e a Reconfigurabilidade. Ambas são

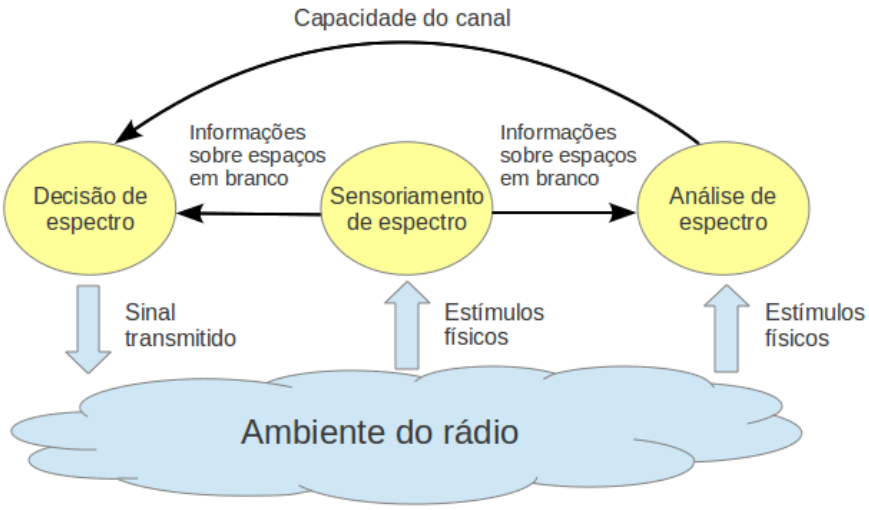


Figura 2.3: *Ciclo cognitivo do dispositivo de rádio.*

descritas em mais detalhes nas seções a seguir.

2.1.1 Capacidade cognitiva

Capacidade cognitiva é a habilidade do dispositivo de capturar ou sensoriar informações sobre o ambiente em que se encontra em tempo real. Para isso, o equipamento deve ser dotado de tecnologia que seja capaz de inspecionar mais de uma frequência e de identificar oportunidades de transmissão que apareçam ao longo do tempo, além de evitar interferência em outros usuários. Através dessa capacidade, as porções do espectro que não são utilizadas podem ser identificadas e consideradas candidatas adequadas para transmissão [4]. A capacidade cognitiva se subdivide em etapas que se inter-relacionam para conferir cognição ao equipamento, são elas: sensoriamento de espectro, análise de espectro e decisão de espectro.

A Figura 2.3 ilustra o ciclo cognitivo e a interação entre suas etapas. O sensoriamento de espectro é a etapa em que o dispositivo monitora o espectro disponível e detecta os espaços em branco. A análise de espectro se encarrega de estimar o tamanho dos buracos detectados na etapa anterior. A última etapa, a decisão de espectro, em que o dispositivo opta por uma taxa de dados, a largura de banda de transmissão e o modo de transmissão e, com isso, efetivamente escolhe a faixa de espectro a transmitir de acordo com as características e os requisitos de usuário.

Uma vez percorrido esse ciclo de cognição, o canal onde irá ocorrer a transmissão é determinado e a comunicação pode ser realizada nessa faixa. Porém, o ambiente pode se modificar no decorrer do tempo e assim, o dispositivo deve acompanhar essas mudanças e, então, a função de Mobilidade de Espectro deve entrar em ação para propor-

cionar continuidade na transmissão [28].

2.1.2 Reconfigurabilidade

A reconfigurabilidade permite o rádio programar dinamicamente seus parâmetros e essa é uma característica que se baseia nas informações coletadas e interpretadas do ambiente [28]. Desse modo, o dispositivo pode ser ajustado em tempo de execução para receber e transmitir em qualquer frequência, sem necessidade de modificação no *hardware*, utilizando diferentes tecnologias. Esses ajustes são desencadeados de acordo com as mudanças ambientais, por exemplo: retomada de transmissão pelo PU, movimentação do usuário ou variação de tráfego.

Existem vários parâmetros que podem ser reconfigurados em um rádio cognitivo, tais como: frequência de operação, modulação e potência de transmissão. Esses itens de reconfiguração estão relacionados ao projeto físico de um dispositivo cognitivo [4].

De acordo com as características percebidas no ambiente, visando aproveitar uma oportunidade de transmissão escolhida, a reconfiguração dos parâmetros pode ser feita de modo isolado ou combinado.

2.2 Conclusão

A solução de Rádios Cognitivos mostra-se como uma abordagem promissora para superar o problema de escassez de espectro, com o intuito de melhorar a eficiência das comunicações. Neste capítulo, foram descritas suas funções e características, com enfoque na capacidade cognitiva e na reconfigurabilidade.

No capítulo seguinte, são apresentadas as técnicas de aprendizado por reforço e estratégias evolutivas, escolhidas por serem técnicas de otimização que reúnem características favoráveis a sua aplicação na DSS.

CAPÍTULO 3

Aprendizado por reforço e estratégias evolutivas

Neste capítulo, são apresentados os conceitos sobre aprendizado por reforço e computação evolucionária com destaque para os dois principais algoritmos de cada uma: *Q-Learning* e Estratégias Evolutivas. Essas abordagens mostram-se candidatas promissoras para solucionar o problema de DSS abordado nesse trabalho.

3.1 Aprendizado por reforço

A história do Aprendizado por Reforço remete ao final de 1979, na Universidade de Massachussets de um dos projetos mais avançados na época, Teoria Heterostática de Sistemas Adaptativos. Esse projeto, desenvolvido por A. Harry Klopft, defendia que redes de neurônios com elementos adaptativos podiam vir a ser uma abordagem promissora para a inteligência artificial adaptativa [52]. A ideia que surgiria no decorrer do trabalho seria a de um sistema de aprendizagem que deseja algo e que adapta o seu comportamento a fim de maximizar uma característica especial do ambiente. Nascia, então, um sistema de aprendizagem hedonista ou nos termos atuais, o aprendizado por reforço.

O grande desafio relacionado ao aprendizado consiste na interpretação entre a experimentação proveniente do ambiente e as ações a serem tomadas de modo a se alcançar o objetivo qualquer que seja a tarefa. Conforme a definição de [44], de modo sucinto, o aprendizado por reforço estuda como um agente autônomo que tem percepção do seu ambiente pode aprender a escolher as melhores ações. Uma vez escolhida a ação a executar, avalia se o objetivo foi atingido através de uma função de recompensa associada a um valor numérico para cada ação tomada em estados distintos.

O aprendizado do agente autônomo ocorre através do mapeamento entre situações e ações produzidas pela interação de tentativa e erro em um ambiente dinâmico. O objetivo do agente é definido usando o conceito de função de recompensa que é a função exata de futuras recompensas, ou reforço, que os agentes procuram maximizar. Em outras palavras, existe uma mapeamento de pares de estado-ação em recompensas que corresponde ao desempenho de uma ação em um dado estado. Assim, diante de uma determinada ação, o agente receberá alguma recompensa na forma de um valor escalar. O

agente aprende as ações que produzem maximização da soma de recompensas quando iniciado em um determinado estado e procedendo ao estado terminal.

A Figura 3.1 representa a interação de um agente com o ambiente e as variáveis envolvidas no processo. Assim, em cada instante de tempo, um agente executa uma ação a_t em algum estado s_t, recebendo um valor numérico de recompensa r_t que indica o valor imediato da transição estado-ação. Uma sequência de estado, ação e recompensa é produzida para seguir uma política que maximize as recompensas esperadas. Para isso, um fator de desconto γ é usado para diminuir exponencialmente o peso de recompensas recebidas no futuro.

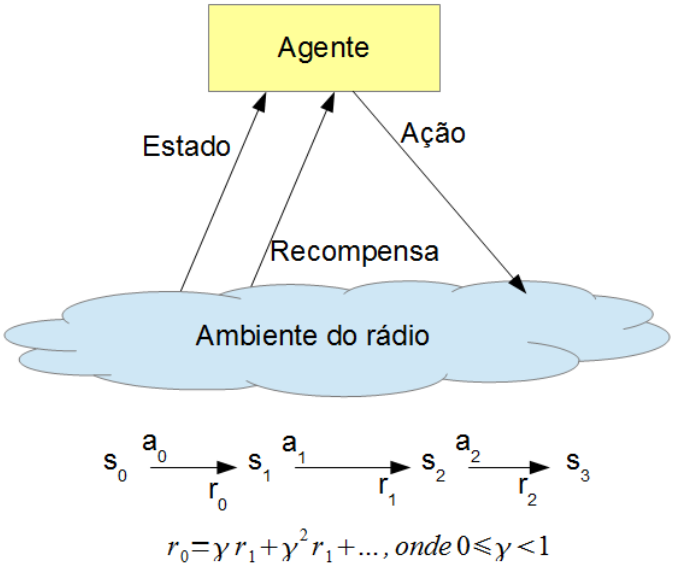


Figura 3.1: *Interação entre o agente, o ambiente e as variáveis envolvidas no processo de aprendizado.*

Os componentes do aprendizado por reforço são: uma política π para tomar uma ação a, em um estado s, uma função de recompensa r e uma função de valoração V que mapeia um estado ao seu valor. A política π define as escolhas do algoritmo de aprendizado e o método de ação a qualquer tempo. A função de recompensa r_t define as relações estado/meta do problema no tempo t. Cada ação ou, mais precisamente, cada par estado-resposta recebe uma medida de recompensa, indicando a vantagem de uma ação para atingir a meta. O algoritmo tem a tarefa de maximizar a recompensa total para atingir sua meta. A função de valoração V é uma propriedade de cada estado do ambiente indicando a recompensa que o sistema pode esperar por futuras ações nesse estado. Diferentemente da função de recompensa que mede a vantagem imediata do par estado-resposta, a função de valoração indica a vantagem a longo prazo do estado no ambiente, conforme descrita pela equação 3-1

$$V^{\pi}(s) = R(s,\pi(s)) + \gamma \sum_{y \in S} P_{sy}(\pi(x)) V^{\pi}(y). \tag{3-1}$$

O uso de aprendizado por reforço é indicado para problemas de otimização e controle, quando não se conhece o modelo do problema e quando se pode treinar com testes e erros. De acordo com [52], existem três diferentes famílias de algoritmos de inferência de aprendizado por reforço: aprendizado por diferença temporal, programação dinâmica, e métodos de Monte Carlo. Toda fundamentação do aprendizado por reforço é baseada no Processo de Decisão de Markov, o qual será comentado em detalhe na próxima seção.

3.1.1 Processo de decisão de Markov

O Processo de Decisão de Markov (MDP – *Markov Decision Process*) é uma forma de modelar processos na qual as transições entre estados são probabilísticas. Em um MDP é possível observar em que estado o processo está e também interferir no processo periodicamente, durante instantes específicos chamados épocas de decisão, executando determinadas ações [47].

Um processo de decisão de Markov é uma tupla (T, S, A, R, P), onde:

1. T é uma sequência de instantes de decisão, no qual cada decisão ocorre em uma época de decisão t, onde $t \in T$. Essa sequência é discreta, podendo ser finita ou infinita.
2. S é um conjunto de estados s, tal que $s \in S$, no qual o processo pode estar em cada instante t. Dentro de um determinado contexto, representa como o ambiente é percebido pelo agente.
3. A é um conjunto de ações a, tal que $a \in A$, e podem ser executadas em diferentes épocas de decisão t. Essa sequência determina a forma de interação do agente com o ambiente.
4. R é uma função que promove a recompensa ou penalização resultante da escolha de uma ação a em um determinado estado s no instante t, denotado por $r_t(s,a)$. Desse modo, representa a percepção que o agente tem sobre o ambiente ao adotar uma determinada ação.
5. P é a probabilidade de transição que determina o estado do próximo instante de decisão $t+1$.

Para especificar uma ação a ser escolhida, é necessário haver algum critério para guiar a decisão e esse critério é chamado de regra de decisão $d_t(s)$. Uma política π é um

conjunto de decisões, tal que $\pi = d_1, d_2, ..., d_N$ e representa uma estratégia de escolha do agente decisor. A política ótima π^* é a sequência de decisões que promovem a otimização das recompensas obtidas no decorrer do processo.

Um Processo de Decisão de Markov caracteriza-se por ser um processo de decisão sequencial em que o conjunto de ações possíveis, retornos e probabilidades de transição são dependentes apenas do estado e da ação corrente, independente do passado. Dessa definição depreende-se a chamada propriedade de Markov, em que o estado resume o passado de forma compacta, sem perder a habilidade de prever o futuro. Sendo assim, é possível predizer qual será o próximo estado e a próxima recompensa esperada dado o estado e ação atuais. No aprendizado por reforço as decisões são tomadas apenas em função do estado atual, satisfazendo portanto essa propriedade [52].

3.1.2 Seleção de ações

A seleção de ações adota métodos que adaptam as estimativas dos valores-ação de acordo com algum critério pré-definido. Nesse sentido, a seleção de ações é classificada de acordo com os critérios adotados, os quais podem ser:

- *Greedy* - busca sempre a escolha daquela ação que promove maiores recompensas.
- ε-*greedy* - escolhe aquela ação que promove maior recompensa considerando um fator ε para condicionar a escolha da próxima ação. Realiza um aproveitamento com uma probabilidade $1 - \varepsilon$, caso contrário, escolhe uma ação aleatória, caracterizando uma exploração. Esse é o método que melhor balanceia a exploração e o aproveitamento.
- *Softmax* - a ação gulosa continua com maior probabilidade, mas as ações restantes tem seus valores ajustados de acordo com estimativas que aplicam a distribuição de Gibbs ou Boltzman. Essa estimativa balanceia a escolha entre uma ação ruim e uma quase ótima.

3.1.3 Caracterização de um problema de aprendizado por reforço

O aprendizado por reforço é definido por ações internas e respostas do ambiente. Assim, os algoritmos de aprendizado por reforço aprendem uma política ótima π^* para atingir uma meta dentro do ambiente. Existem pontos chaves que o caracterizam e ajuda a atingir essa política ótima, conforme a seguir.

1. Tentativa e erro - como não existe uma etapa de treinamento, o aprendizado é produzido pela experimentação do agente no ambiente e isso envolve tentar, errar ou acertar, promovendo um ciclo de retroalimentação do mecanismo de aprendizado. Isso é necessário porque o ambiente real possui um comportamento estocástico.

2. Recompensa tardia - como não existe a fase de treinamento, o agente produz uma sequência de valores de recompensa imediata ao executar determinada ação. O agente pode, inicialmente, sacrificar as recompensas imediatas em função do aumento da recompensa de longo prazo, que serão favorecidas pelo aprendizado acumulado.
3. Exploração *versus* aproveitamento - a estratégia a ser utilizada na experimentação pode influenciar para onde o aprendizado caminhará. Por isso, existe o dilema de qual estratégia escolher: explorar estados e ações desconhecidos ou aproveitar estados e ações já conhecidos. A primeira estratégia tem o objetivo de reunir novas informações, arriscando o valor de recompensa que será obtida, mas ampliando as alternativas futuras. Já a segunda tem possiblidade de resultar em alta recompensa pelo fato de ser uma ação conhecida, contribuindo para maximizar a recompensa acumulada. Não se pode explorar o tempo todo, nem aproveitar sempre, deve haver alternância entre as duas estratégias.

3.1.4 *Q-Learning*

Um dos métodos mais importantes de aprendizado por reforço é o *Q-Learning* [58], uma variante da abordagem de diferença temporal, onde Q é uma função que mapeia pares estado-ação em valores aprendidos, Q : estado-ação $\rightarrow$ valor. A cada instante de tempo t de aprendizado, uma tabela denominada tabela-Q é atualizada para cada estado-ação com um valor denominado valor-Q, que estima o nível de recompensa para uma ação. Assim, mudanças nos valores-Q implicam em mudanças na ação do algoritmo.

Uma importante característica desse algoritmo é que sua convergência para valores ótimos de Q não é dependente da política que está sendo utilizada. De modo geral, o ciclo de cognição desempenhado pelo *Q-Learning* é exibido na Figura 3.2 na forma de fluxograma. Nesse fluxograma, estão as cinco etapas: observação, planejamento, decisão, ação e aprendizado. A observação do estado está relacionada ao conhecimento sobre o ambiente em que o dispositivo se encontra. Diante dessa observação, é necessário planejar uma escolha: explorar ou aproveitar. Ao executar a ação escolhida, o agente deve novamente interagir com o ambiente, calcular a recompensa e atualizar a tabela-Q.

A função-valor Q se aproxima diretamente da função-valor ótima, Q^*, através da atualização da tabela-Q à medida que o estado-ação é visitado. Para atualizar os valores da tabela, é utilizada a seguinte equação:

$$Q_{t+1}(a) \leftarrow (1-\alpha)Q_t(a) + \alpha r_{t+1}(a), \tag{3-2}$$

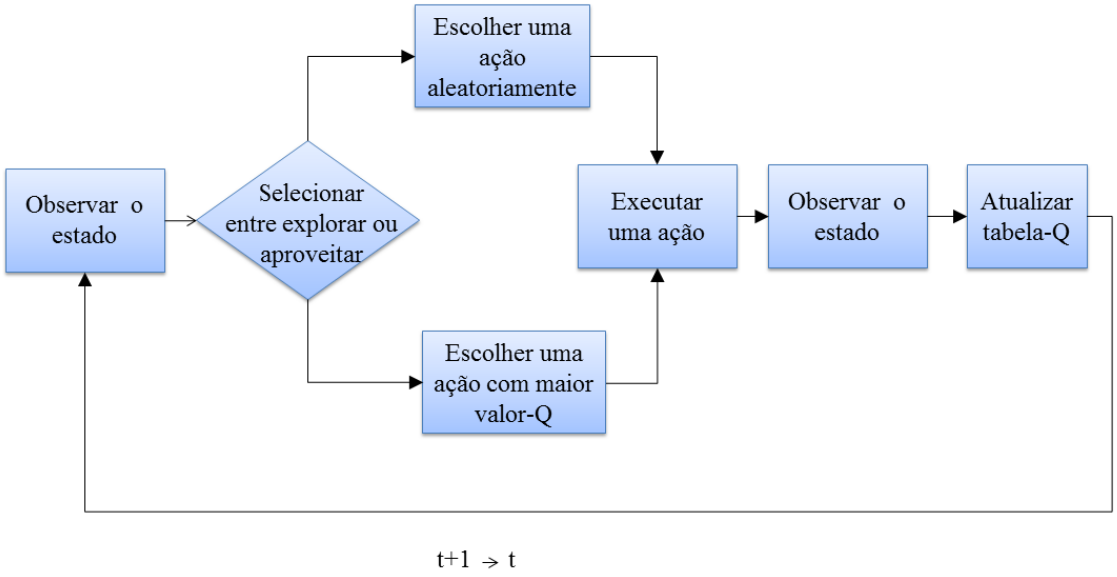


Figura 3.2: *Ciclo cognitivo do Q-Learning.*

onde a é a ação escolhida, $Q_{t+1}(a)$ corresponde ao valor-Q atual para uma ação a; $Q_t(a)$ é o valor-Q calculado na época de decisão anterior para a ação a; α é a taxa de aprendizado, definida no intervalo $0 \leq \alpha \leq 1$ e $r_{t+1}(a)$ é a recompensa obtida na época de decisão $t+1$ ao adotar uma ação a.

O método do *Q-Learning* foi o primeiro que indicou fortes provas de convergência. Watkins mostrou que se cada estado-ação fosse visitado infinitas vezes, a função-valor Q iria convergir com probabilidade de 1 para Q^*, usando α suficientemente pequeno. O valor-Q ótimo é extraído de uma escolha ótima das ações tomadas de acordo com a política de seleção de ações

$$a^*(s) = argmax_a Q(s,a), \tag{3-3}$$

onde $a^*(s)$ representa a ação ótima no estado s escolhida de acordo com o maior valor-Q encontrado na tabela-Q, representado na equação por $argmax_a Q(s,a)$.

Uma ação é tomada de acordo com a política $\varepsilon - greedy$

$$\pi(s) = \begin{cases} a^* & \text{com probabilidade } 1-\varepsilon \\ a_{aleatória} & \text{com probabilidade } \varepsilon, \end{cases} \tag{3-4}$$

onde ε representa o fator de desconto que condiciona a escolha da próxima ação a^*, aproveitamento ou $a_{aleatória}$, exploração.

Uma avaliação sobre o algoritmo *Q-Learning* utilizado neste trabalho e sua adequação ao problema de seleção dinâmica de espectro serão apresentados na Seção 5.

3.2 Computação evolutiva

A característica evolutiva é baseada na seleção natural, ou competição inter-membros, cuja teoria foi formulada por Charles Darwin. Nesse contexto, somente os indivíduos considerados melhores possuem maior probabilidade de adaptar-se e sobreviver. Como consequência, seu material genético será propagado e com alta probabilidade de que seus descendentes sejam mais fortes e resistentes [33].

Inspirando-se nesse processo biológico de evolução e abstraindo esses conceitos para o contexto computacional, surge o conceito de Computação Evolutiva. Esse é um ramo da Inteligência Artificial com objetivos de otimização por combinação [17, 19]. Possui aplicações bem sucedidas em diversas área como: programação automática, processamento de sinais, bioinformática e sistemas sociais.

A Computação Evolutiva compreende uma variedade de algoritmos aplicados à solução de problemas de otimização. Esses algoritmos compartilham o princípio da evolução natural em que as pressões do ambiente sobre uma dada população causam seleções naturais, ou seja, a sobrevivência dos mais aptos.

Na Computação Evolutiva, o ambiente é representado por uma função de qualidade que se deseja otimizar, a população representa um conjunto de soluções candidatas escolhidas aleatoriamente e há uma função de qualidade para mensurar a aptidão de um indivíduo, ou solução. Essa função de qualidade é utilizada como métrica para escolher os genitores da próxima geração, a qual é criada através da recombinação de dois ou mais indivíduos ou ainda, a mutação de um único indivíduo. Diante desse comportamento, duas forças são fundamentais para os sistemas evolucionários: variabilidade e seleção. A primeira cria a diversidade necessária para explorar o espaço de soluções, enquanto a segunda impulsiona a qualidade da solução [6].

Em muitos problemas com os quais nos deparamos, a função matemática que o modela não é conhecida, o espaço de busca é grande e os valores de certos parâmetros são obtidos a partir de simulações. Problemas com essas características são fortes candidatos a usufruir da Computação Evolutiva para buscar uma solução aproximada da ótima. Um bom exemplo disso é o problema da Mochila Multidimensional [37].

A Computação Evolutiva abrange de forma ampla, as seguintes áreas: Algoritmos Genéticos, Programação Genética, Programação Evolutiva e Estratégia Evolutiva. Essas áreas possuem uma mesma base conceitual que envolve: simular a evolução de estruturas individuais através de processos de seleção, recombinação, mutação e reprodução, produzindo as melhores soluções. A seção a seguir descreve a Estratégia Evolutiva. As demais abordagens citadas são descritas no Apêndice A.

3.2.1 Estratégias evolutivas

Estratégia Evolutiva (ES – *Evolution Strategy*) [49] é uma classe de algoritmos da Computação Evolucionária e surgiu para solucionar problemas de otimização de parâmetros contínuos ou discretos. Foi proposto por I. Rechenberg e H. P. Schwefel, na Alemanha, com a publicação de um livro em 1973[1] que fundamenta a Estratégia Evolutiva [22].

A característica principal da ES é a adaptação *online* dos parâmetros que regem o processo evolutivo. Como consequência, um algoritmo baseado em ES executa simultaneamente duas tarefas: a solução de um problema de otimização específico e a própria calibragem do algoritmo para resolver o problema em questão.

A Estratégia Evolutiva é a que mais se aproxima da teoria evolutiva de Lamarck. Essa teoria, publicada em [38], defende que as variações no meio ambiente levam o indivíduo a se adaptar buscando a perfeição, seguindo duas leis: a do uso e desuso e a da transmissão dos caracteres adquiridos. Nesse contexto, a Estratégia Evolutiva utiliza como operações a mutação auto-adaptativa e a recombinação com operador de seleção determinístico no processo de busca da solução.

Os algoritmos de Estratégia Evolutiva trabalham com quantidade diferente de indivíduos na população e no conjunto a partir do qual a próxima geração é criada. Essa é justamente a característica que diferencia os tipos de estratégias evolutivas [7], listadas a seguir.

1. Dois-membros ou (1 + 1)-EE - composta por um único indivíduo, e apenas o operador genético de mutação é utilizado.
2. Multimembros ou (μ+ 1)-EE - método de gradiente probabilístico, onde a perturbação é Gaussiana e adaptativa.
3. Multimembros ou (μ+ λ)-EE - seleção opera no conjunto união de pais e filhos.
4. Multimembros (μ, λ)-EE - seleção somente dos, os pais são eliminados.

Entre esses tipos de estratégias existentes, a de Dois-Membros é a que reune características que se adequam ao problema de DSS, e por isso será apresentada em detalhe. A população é composta por um único indivíduo genitor e um único descendente, sendo esse gerado exclusivamente através de um operador de mutação. Eles competem entre si pelo papel de genitor da próxima geração. De maneira formal, a estratégia (1+1)-EE pode ser descrita pela tupla:

$$\text{(1+1)-EE} = (P^0, m, s, h, f, t), \tag{3-5}$$

[1] "Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution", Frommann-Holzboog Verlag, 1973

onde $P^0 = (x^0, \sigma^0) \in I,\ I = \mathbb{R}^n \times \mathbb{R}^n$, denota a população inicial; $m : I \to I$ é o operador de mutação; $s : I \times I \to I$ é o operador de seleção; $h : \mathbb{R}^n \to \mathbb{R}^n$ denota uma heurística que controla a variação do operador de mutação; $f : \mathbb{R}^n \to \mathbb{R}$ é a função objetivo; e $t : I \times I \to \{0,1\}$ denota um critério de parada.

Um par de vetores reais (x, σ) representa um indivíduo da população. O vetor x representa um ponto no espaço de soluções, sendo que cada componente de x corresponde a um parâmetro a ser otimizado. σ é um vetor de desvio padrão a ser utilizado na mutação de x. No início do processo evolutivo, a população inicial P^0 consiste de um único indivíduo, (x^0, σ^0), que produz por mutação um único descendente, resultando em uma população P'^t, representada nas Equações 3-6, 3-7 e 3-8:

$$P'^t = (a'^t_1, a'^t_2) \in I \times I, \tag{3-6}$$

$$a'^t_1 = P^t = (x^t, \sigma^t), \tag{3-7}$$

$$a'^t_2 = m(P^t) = (x'^t, \sigma^t). \tag{3-8}$$

O operador de seleção, então, escolhe o indivíduo mais apto do conjunto formado pelo genitor e seu descendente. Esse indivíduo se tornará o genitor da próxima geração. Para problemas de maximização, a operação de seleção pode ser descrita pela Equação 3-9:

$$P^{t+1} = s(P'^t) = \begin{cases} a'^t_2 & \text{se } f(x'^t) \geq f(x^t) \\ a'^t_1 & \text{caso contrário.} \end{cases} \tag{3-9}$$

O processo de iteração $P^t \to P^{t+1}$ termina quando a condição $t(a'^t_1, a'^t_2) = 1$ é satisfeita. Na prática, t é uma função que depende da implementação. A variação da mutação é feita de forma *online*, a cada k gerações, pela aplicação de uma heurística h. Essa heurística permite aumentar ou diminuir o nível de variabilidade da mutação ao longo do processo evolutivo. Em geral, o início do processo de busca requer variações maiores. Essas variações, no entanto, tendem a diminuir à medida que a busca se torna mais localizada, em virtude das informações obtidas com as iterações anteriores.

A estrutura de um algoritmo EE é simples e independente do tipo:

1. uma população inicial é formada e seus indivíduos são avaliados;
2. os melhores indivíduos da população sobrevivem e são selecionados para se tornarem os indivíduos pais da próxima geração;
3. os parâmetros estratégicos são atualizados;
4. através de recombinação de dois ou mais indivíduos pais (ou mutação, de um único indivíduo), novos indivíduos (descendentes) são criados e avaliados;

5. repetição do processo a partir do item 2 até que algum critério seja alcançado, levando ao término da execução.

A Figura 3.3 ilustra de modo esquemático o funcionamento dos algoritmos de Estratégia Evolutiva.

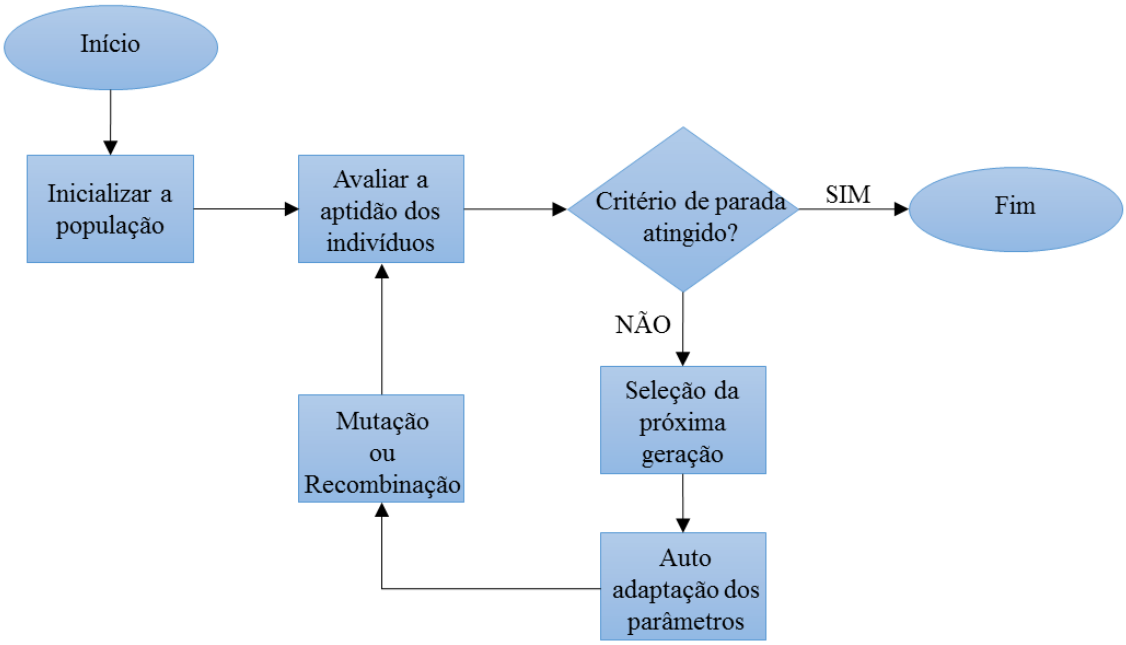


Figura 3.3: *Fluxograma de um algoritmo clássico de estratégia evolutiva.*

3.3 Conclusão

Neste capítulo, foram apresentadas técnicas que promovem a resolução de problemas de otimização com fundamentos em Aprendizado de Máquina e Inteligência Artificial. Foram também destacadas aquelas características de cada técnica que favorecem a resolução do problema de DSS. O capítulo seguinte apresenta o modelo para o problema e as etapas de evolução do novo algoritmo proposto.

CAPÍTULO 4

Modelagem do problema de DSS e algoritmos propostos

Conforme descrito no Capítulo 2, o problema de DSS em rádios cognitivos é um dos pontos chaves para a tecnologia. A resolução desse problema com foco na qualidade de serviço pode proporcionar aos RCs o alcance do seu objetivo tecnológico: aproveitamento oportunístico e eficiência no aproveitamento do espectro.

Com essa motivação, este capítulo descreve um modelo que representa o problema e a proposição de novos algoritmos inspirados em Estratégia Evolutiva para promover a otimização do mecanismo de DSS.

4.1 Modelo proposto

O modelo proposto consiste em uma divisão de espectro em que o problema de DSS é descrito por uma representação compacta dos canais. Embora seja um modelo simplificado, são feitas considerações importantes tais como: características do canal e o custo de troca de canais.

Seja $C = \{c_0, c_1, ..., c_n\}$ o conjunto de canais não sobrepostos, em que cada canal ou frequência c_i possui a mesma largura de banda. A vazão efetiva de cada canal c_i varia no decorrer do tempo e as condições dos canais são representadas pela variável aleatória X_{ij}. Essa variável resume as características do meio, tais como perda de pacotes e padrão de transmissão do PU. Como a avaliação deste trabalho possui foco na tentativa de transmissão, que pode tanto ser bem sucedida quanto falhar, foi empregada a representação discretizada do tempo. Em cada instante de tempo T_j, um par SU (transmissor e receptor) escolhe um canal c_i para enviar pacote. Além disso, um SU é capaz de sensoriar apenas um único canal a cada instante T_j.

A Figura 4.1 ilustra o modelo descrito para um cenário com $n+1$ canais durante $m+1$ instantes de tempo.

Algumas premissas relacionadas à possibilidade de colisão e interferência ao PU foram adotadas no modelo. Desse modo, antes de transmitir, um SU sensoria o

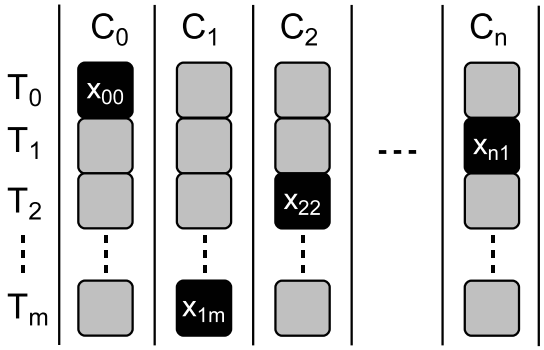


Figura 4.1: *Modelo proposto para representação dos canais no decorrer do tempo.*

meio para evitar colisões, particularmente com os PUs. Contudo, um PU pode iniciar a transmissão enquanto um SU ainda está transmitindo. Nesse caso, a interferência com o PU é minimizada pelo tamanho do pacote do SU, o qual ocupa o meio por um período muito curto de tempo em relação à média de ocupação das transmissões do PU. Os SUs são fontes saturadas de tráfego com o intuito de mensurar a capacidade máxima disponível de cada canal, ou seja, o transmissor SU sempre tem dados a transmitir.

Também foi assumido que um par SU é capaz de estabelecer coordenação em um canal de controle, separado do canal de dados, pois o modelo proposto é focado em canais de dados. Como a troca de canais envolve uma certa sobrecarga de controle, o tempo para transmissão de dados deve ser reduzido quando essa operação ocorre. De acordo com [63], a troca de canal leva quase a mesma quantidade de tempo de um pacote de dados do SU. Neste trabalho, foi adotado uma abordagem similar, mas foi aproximada para a duração de um pacote completo. Assim, depois de uma troca de canal, o par SU não transmite ou deixa de competir durante o tempo de uma tentativa de transmissão.

4.2 Algoritmo proposto para a DSS e sua evolução

O algoritmo de DSS proposto neste trabalho foi inspirado nos fundamentos de Estratégia Evolutiva apresentados na Seção 3.2.1. Conforme comentado previamente, o algoritmo proposto foi baseado em uma estratégia (1+1)-ES. Foi definido que um indivíduo representaria um canal de dados e a função objetivo mediria a vazão obtida pelo canal dentro de uma época de decisão. A maior vazão determina o melhor indivíduo, sendo o objetivo principal a maximização da vazão global. No final de uma época de decisão, uma nova geração é criada com um único pai que gera um único descendente por meio de mutação.

A estratégia (1+1)-ES foi escolhida para uso no problema de DSS porque, a cada instante, o SU é capaz de observar apenas o canal que escolheu para transmitir. Assim, apenas um indivíduo pode ser criado a cada geração, resultando em uma população

formada por um único genitor e um único descendente. O processo de seleção consiste na escolha do indivíduo mais apto, ou seja, o canal que exibiu maior número de sucessos em suas transmissões. Assim, o melhor canal é o que será usado na próxima época de decisão. Contudo, diferente da abordagem ES tradicional, o algoritmo proposto não elimina os indivíduos de pior desempenho. Como resultado, em cada geração, a população compara um pai e λ descendentes ou canais onde $\lambda < n+1$, onde n é a quantidade de canais.

A decisão de não eliminar o indivíduo menos apto permite que esses canais sejam revisitados no futuro. Isso é essencial no contexto de seleção dinâmica de canais, pois as habilidades dos indivíduos em diferentes gerações podem mudar. No problema de DSS, tipicamente, as condições dos canais podem variar de maneira significativa ao longo do tempo, ou seja, o nível de aptidão dos indivíduos pode se alterar em diferentes gerações.

O projeto do algoritmo levou em consideração, inicialmente, um par SU comunicante para depois evoluir para múltiplos pares SU. Assim, a primeira versão do algoritmo projetado recebeu o nome de ES^3(*Evolution Strategy for Spectrum Selection*). A avaliação com apenas um par SU comunicante apresentou resultados promissores [9], superando o *Q-Learning* na maior parte dos cenários analisados [8]. Para avaliar múltiplos pares de SUs foram necessárias algumas adaptações no algoritmo inicial de modo que pudesse lidar com a concorrência entre os pares comunicantes diante de uma oportunidade de transmissão. As melhorias aplicadas ao algoritmo original implicaram em uma nova versão que recebeu o nome de E^{2}S^3 (*Enhanced Evolution Strategy for Spectrum Selection*). As duas versões do algoritmo são apresentadas em detalhe nas próximas seções.

4.2.1 *Evolution Strategy for Spectrum Selection* - ES^3

O pseudocódigo para o ES^3 é apresentado no Algoritmo 4.1 e comentado a seguir.

Seja $x \in \mathbb{R}$ um número gerado por uma distribuição Normal de média zero e desvio padrão σ denotada por $N(0,\sigma)$. Seja $\phi : \mathbb{R} \rightarrow C$ uma função que mapeia x em um canal $c_i \in C$. Assim, ϕ é uma função que discretiza intervalos da distribuição Normal em canais, como mostrado na Figura 4.2. Nessa discretização, valores distantes até $\pm 3\sigma$ da média são agrupados em $2(n+1)$ intervalos, de forma que o i-ésimo intervalo à esquerda da média tem a mesma largura do i-ésimo intervalo à direita. Valores fora do intervalo $[-3\sigma, +3\sigma]$ são atribuídos a c_0, o que ocorre com probabilidade de $0,3\%$. Um par de intervalos simétricos, ou seja, equidistantes da média, representa um canal c_i.

Seja $V = \{v_0, v_1, ..., v_n\}$ o conjunto que representa os pares de intervalos simétricos da discretização de $N(0,\sigma)$. Cada v_j é uma tupla na forma $(lie, lse, lid, lsd, c, f(c))$, onde lie e lse são, respectivamente, os limites inferior e superior do intervalo à esquerda da média; lid e lsd são os limites inferior e superior do intervalo à direita da média; $c \in C$

Algoritmo 4.1: ES^3

Entrada: t_D, σ, k

$tamIntervalo \leftarrow 3/(n+1)$
para $i = 0 \rightarrow n$ **faça**
  //Atribui um canal aleatório ao intervalo $v_i \in V$
  $v_i[c] \leftarrow random(0,n)$
  //Gera os valores (lie, lse, lid, lsd) para v_i
  $geraIntervalo(v_i, tamIntervalo)$
  enquanto *houver transmissão* **faça**
    para $l = 1 \rightarrow k$ **faça**
      $x \leftarrow \text{Normal}(0,\sigma)$
        //segundo Equação 4-1
      $canal \leftarrow \phi(x)$
      $N_D \leftarrow 0$
      para $j = 1 \rightarrow t_D$ **faça**
        $sucesso \leftarrow transmitaEm(canal)$
        se $sucesso = True$ **então**
          $N_D \leftarrow N_D + 1$
          $f(c_i) = N(D)$
        fim
      fim
      //Ordena $C = (c_i, f(c_i))$ decrescentemente por $f(c_i)$
      $ordena(C)$
      para $i = 0 \rightarrow n$ **faça**
        $v_i[c] \leftarrow C_i$
        se $l = k$ **então**
          //Redimensiona os intervalos de cada canal
          $tamIntervalo \leftarrow (f(c_i)/\sum f(c_i)) \times 3$
          $geraIntervalo(v_i, tamIntervalo)$
        fim
      fim
    fim
  fim
fim

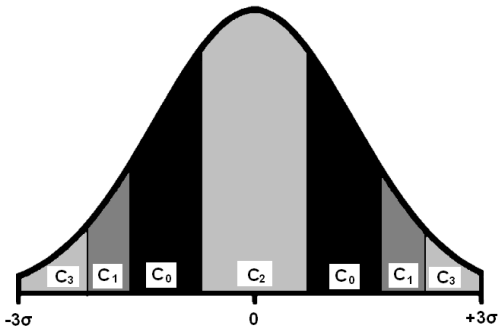


Figura 4.2: *Discretização de uma distribuição $N(0,\sigma)$ em 4 canais no ES^3.*

é o canal representado pelo par de intervalos e $f(c)$ é a vazão obtida no canal c. O conjunto V é ordenado de forma que o j-ésimo par de intervalos com maior probabilidade ocupe a j-ésima posição. A função ϕ é definida da seguinte forma:

$$\phi(x) = \begin{cases} c_0 & \text{, se } x \notin [-3\sigma, +3\sigma] \\ v_j[c] & \text{, se } x \in [v_j[lie], v_j[lse]] \vee x \in [v_j[lid], v_j[lsd]]. \end{cases} \quad (4\text{-}1)$$

Aplicar a estratégia (1+1)-ES ao problema de DSS consiste em definir os indivíduos, os operadores de mutação e seleção e a heurística para controlar a variação da mutação no contexto desse problema.

No ES^3, cada indivíduo é modelado como um par $(x, \phi(x))$, onde x é um número real e ϕ é a função definida na Equação 4-1. A população inicial é formada por um único canal $c^0 = (x^0, \phi(x^0)) \in C$, escolhido aleatoriamente. O operador de mutação é representado pela distribuição $N(0,\sigma)$. A cada geração, um novo indivíduo é criado através do sorteio de um número x que segue a distribuição $N(0,\sigma)$. O número x sorteado é então mapeado para um canal usando a função ϕ.

Uma questão essencial que surge nesse modelo é como atribuir um canal $c_i \in C$ a uma tupla v_j. No ES^3, isso é feito atribuindo os canais com maior vazão aos intervalos com maior probabilidade. Essa política assegura chances maiores para os indivíduos mais aptos, ou seja, os canais que possibilitam as maiores vazões.

Sempre que um canal c_i é sorteado pelo processo de mutação, esse canal é visitado e t_D tentativas de transmissão são efetuadas nesse canal, resultando em uma vazão $f(c_i)$ igual ao número de transmissões bem sucedidas. A operação de seleção então posiciona o canal c_i numa tupla v_j de acordo o nível de vazão obtido em $f(c_i)$. De fato, no ES^3, o operador de seleção posiciona os canais em ordem decrescente de vazão, resultando em um conjunto $C' = \{c'_0, ..., c'_n\}, c'_i \in C$, e atribui o conjunto formado de modo que o i-ésimo canal de C' seja atribuído para a i-ésima tupla de V. Esse processo evolutivo pelo qual os canais passam se repete por k gerações, ao fim das quais os limites dos intervalos de discretização de $N(0,\sigma)$ são ajustados automaticamente.

O reajuste visa aumentar ou diminuir a largura de cada par de intervalos de maneira proporcional ao desempenho do último canal representado por aquele par, de acordo com a Equação 4-2.

$$tamIntervalo \leftarrow (f(c_i)/\sum f(c_i)) \times 3 \tag{4-2}$$

Essa política tem impacto nas chances de sorteio dos próximos canais representados pelo par de intervalos das próximas k gerações. Na Figura 4.3, para exemplificar, é apresentada uma saída ao final de uma época para uma execução do ES^3. Nela temos os cálculos dos intervalos para cada canal, conforme a Equação 4-2 e os limites que representam os canais na Gaussiana de acordo com os cálculos obtidos.

```
**********fim de uma geração
{'num_canal': 0, 'tam_intervalo': 1.0, 'vazao': 0.8666666666666667}
{'num_canal': 1, 'tam_intervalo': 0.5, 'vazao': 0.20000000000000001}
{'num_canal': 2, 'tam_intervalo': 1.5, 'vazao': 0.6333333333333333}

Gaussiana:
[{'C': 2, 'lse': -1.5, 'lie': 0, 'lid': 0, 'lsd': 1.5},
 {'C': 0, 'lse':-2.5, 'lie':-1.5 'lid': 1.5, 'lsd': 2.5},
 {'C': 1, 'lse':-3.0, 'lie':-2.5, 'lid': 2.5, 'lsd': 3.0}]
```

Figura 4.3: *Intervalos e Representação da Gaussiana ao fim de uma geração.*

Ao realizar esse ajuste nos intervalos de cada canal, o algoritmo proporciona um ajuste automático entre as estratégias de aproveitamento e de exploração conforme a necessidade observada no ambiente. Isso é essencial em ambientes dinâmicos para proporcionar reação rápida.

4.2.2 *Enhanced Evolution Strategy for Spectrum Selection* - E^{2}S^3

O E^{2}S^3 é uma versão melhorada do ES^3, que caracteriza-se pela auto-adaptação, alcançada pela inclusão da duração da época de decisão (e_D) como um parâmetro estratégico. Enquanto no ES^3 a época de decisão tinha uma duração fixa, nessa nova versão, este parâmetro é tratado como uma parâmetro variável dentro do processo evolutivo. O pseudocódigo para o E^{2}S^3 é apresentado no Algoritmo 4.2 e seus detalhes são discutidos a seguir.

No E^{2}S^3 também foi utilizada uma distribuição Normal com média zero e desvio padrão σ com notação $N(0,\sigma)$. Porém, existe uma diferença, enquanto no ES^3 a curva completa foi utilizada, no E^{2}S^3 apenas a porção positiva foi utilizada. Com essa decisão de

Algoritmo 4.2: E^2S^3

Entrada: ${e_D}^0$, σ, C

1 $larguraIntervalos \leftarrow 3\sigma/(n+1)$
2 $troca \leftarrow$ Falso
3 $canalAnterior \leftarrow -1$
4 **para** $i = 0 \rightarrow n$ **faça**
5 `//Atribui um canal aleatório ao intervalo` $v_i \in V$ `e calcula seus limites` $v_i[c] \leftarrow random(0,n)$
6 $calculaLimiteIntervalos(v_i, larguraIntervalos)$
7 **fim**
8 **enquanto** *houver transmissões* **faça**
9 $x \leftarrow \text{Normal}(0,\sigma)$
10 $canalAtual \leftarrow \phi(x)$
11 `//Conforme Equação 4-3`
12 **se** $canalAnterior \neq -1$ **então**
13 **se** $canalAnterior \neq canalAtual$ **então**
14 $troca \leftarrow$ Verdadeiro
15 **fim**
16 $e_D \leftarrow calcularDuracaoNovaEpoca(e_D, troca)$
17 **senão**
18 $e_D = {e_D}^0$
19 **fim**
20 $N_D \leftarrow 0$
21 **para** $j = 1 \rightarrow e_D$ **faça**
22 **se** $troca = Verdadeiro$ **então**
23 $sucesso \leftarrow$ Falso
24 $troca \leftarrow$ Falso
25 **senão**
26 $sucesso \leftarrow transmitirEm(canalAtual)$
27 **fim**
28 **se** $sucesso = Verdadeiro$ **então**
29 $N_D \leftarrow N_D + 1$
30 **fim**
31 **fim**
32 $f(canalAtual) \leftarrow N_D/e_D$
33 $ordena(C)$ `//Ordena` $C = (c_i, f(c_i))$ `em ordem decrescente de` $f(c_i)$
34 **para** $i = 0 \rightarrow n$ **faça**
35 $v_i[c] \leftarrow c_i$
36 $larguraIntervalo \leftarrow (f(c_i)/\sum f(c_i)) \times 3\sigma$
37 $calculaLimiteIntervalos(v_i, larguraIntervalo)$
38 **fim**
39 $canalAnterior \leftarrow canalAtual$
40 **fim**

projeto, a representação dos intervalos possui metade dos parâmetros. A diminuição dos intervalos para cada canal foi a motivação para o uso da porção positiva da curva Normal, pois diminuiria os cálculos e não implicaria prejuízos para a função de mapeamento.

Seja $\phi : \mathbb{R} \rightarrow C$ uma função que mapeia x, um número aleatório gerado por $N(0,\sigma)$, em um canal $c_i \in C$. Como ilustrado na Figura 4.4, a função ϕ discretiza valores contínuos de uma distribuição $N(0,\sigma)$, que é até no máximo 3σ da média, em $(n+1)$ intervalos, em que cada um representa um canal c_i. Seja $V = \{v_0, v_1, ..., v_n\}$ um conjunto de intervalos discretizados da distribuição $N(0,\sigma)$. Cada intervalo $v_j \in V$ é uma tupla $(li, ls, c, f(c))$, onde li e ls são respectivamente limite inferior e superior dos intervalos, $c \in C$ é o canal representado pelo intervalo e $f(c)$ é a vazão alcançada pelo canal c.

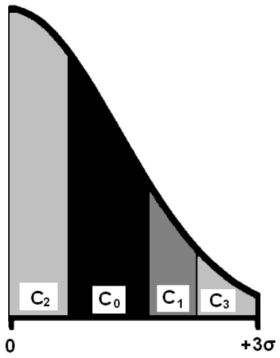


Figura 4.4: *Discretização de uma distribuição $N(0,\sigma)$ em 4 canais no E^2S^3.*

Assim como no ES^3, o conjunto V é ordenado de modo decrescente de probabilidade de acordo com a largura dos intervalos. Os canais com vazão alta são associados a intervalos com maior probabilidade. Essa política assegura maior chance para os melhores indivíduos, ou seja, os canais que oferecem melhores resultados. A função ϕ é definida como:

$$\phi(x) = \begin{cases} c_0 & \text{, se } |x| \notin [0, +3\sigma] \\ v_j[c] & \text{, se } |x| \in [v_j[li], v_j[ls]]. \end{cases} \quad (4\text{-}3)$$

De modo diferente do ES^3, no E^{2}S^3 cada indivíduo é representado por um par (x^e, δ^e), onde x^e é um número aleatório da distribuição normal e $\delta^e = (V^e, {e_D}^e)$ é o vetor de parâmetros estratégicos. A diferença consiste justamente no acréscimo dos parâmetros estratégicos à representação do indivíduo e é o que proporcionará a característica de auto-adaptação a essa versão do algoritmo. A população inicial consiste de um único canal $c^0 = (x^0, (V^0, {e_D}^0)) \in C$ que foi escolhido aleatoriamente. O operador de mutação é representado pela distribuição $N(0,\sigma)$. Ao final de uma época de decisão, um novo

indivíduo é criado pelo sorteio de um número aleatório x através de uma distribuição $N(0,\sigma)$.

O valor x que representa o indivíduo é, então, mapeado em um canal c_i por meio da Equação 4-3. O canal c_i é visitado e ${e_D}^e$ tentativas de transmissão são realizadas nesse canal, resultando em uma vazão $f(c_i) = N_D/{e_D}^e$, onde N_D é a quantidade de transmissões bem sucedidas. Em seguida, o operador de seleção reposiciona o canal c_i em um intervalo v_j de acordo com o valor de vazão de $f(c_i)$. De fato, o operador de seleção ordena os canais pela função objetivo em ordem decrescente e associa o i-ésimo canal de um conjunto ordenado para a i-ésima tupla de V. Como parte do processo de adaptação, os limites de intervalo de V são reajustados para aumentar ou diminuir a largura do intervalo proporcionalmente a vazão obtida em cada canal. Esse procedimento afeta as chances de cada canal nas próximas gerações. Finalmente, a duração da próxima época de decisão, ${e_D}^{e+1}$, é ajustada, para assegurar reação rápida às mudanças no ambiente, de acordo com aumento ou diminuição da vazão na época avaliada.

O ajuste da época de decisão futura é dependente do desempenho obtido na época anterior. A Tabela 4.1 resume o comportamento do ajuste da época de decisão. Nela, a época atual (${e_D}^{e+1}$) é comparada com a época anterior (${e_D}^e$). Também são comparadas a vazão atual ($\sum f(c_i)^{e+1}$) com a vazão anterior ($\sum f(c_i)^e$) para então decidir entre incrementar ou decrementar a duração da próxima época de decisão.

Tabela 4.1: *Adaptação da época de decisão.*

Época de decisão	Vazão	Incremento
${e_D}^{e+1} > {e_D}^e$	$\sum f(c_i)^{e+1} > \sum f(c_i)^e$	+5
	$\sum f(c_i)^{e+1} < \sum f(c_i)^e$	-5
${e_D}^{e+1} < {e_D}^e$	$\sum f(c_i)^{e+1} > \sum f(c_i)^e$	-5
	$\sum f(c_i)^{e+1} < \sum f(c_i)^e$	+5
${e_D}^{e+1} = {e_D}^e$	$\sum f(c_i)^{e+1} \leq \sum f(c_i)^e$	$\pm$5 conforme sorteio

A adaptação da época de decisão é a característica que faz o E^{2}S^3 ser uma evolução do ES^3, pois ao recalcular a próxima época de transmissão de acordo com o desemepenho, o algoritmo consegue se auto-calibrar de acordo com os estímulos observados no ambiente.

4.3 Conclusão

Este capítulo apresentou uma modelagem para o problema investigado baseada em processo estocástico e as versões dos algoritmos propostos neste trabalho: ES^2 e E^{2}S^3. O projeto de ambos algoritmo foi detalhado e discutida a necessidade de melhoria na

versão inicial, para atender necessidades em ambientes de maior concorrência entre os dispositivos.

A partir dessas considerações, o próximo capítulo traz a descrição dos cenários de simulação e detalhes do simulador que implementa os mecanismos descritos neste capítulo.

CAPÍTULO 5

Avaliação dos mecanismos e análise de resultados

Neste capítulo, são descritos os cenários, a plataforma de simulação e a versão do *Q-Learning*. Além disso, um estudo sobre a influência dos parâmetros do *Q-Learning* é apresentado. Os mecanismos propostos neste trabalho e a versão do *Q-Learning* são avaliados na plataforma de simulação e os resultados obtidos em diversos contextos são analisados e comentados.

5.1 Cenários de simulação

Para as execuções foram utilizados quatro cenários diferentes, classificados em dois tipos: estático e dinâmico. Em todos os cenários, a quantidade de canais varia de 3 a 12. Esse intervalo representa SUs com maior e menor capacidade de seleção, respectivamente.

No cenário estático, a chance de sucesso das transmissões em cada canal é definida através de uma distribuição Uniforme U com intervalo de sorteio fixo $[a,b]$ ao longo de toda a simulação. A vantagem da distribuição Uniforme é a possibilidade de criar cenários com algum nível de controle, mantendo a aleatoriedade inerente a ambientes sem fio. Desse modo, é possível definir facilmente um intervalo de sorteio $[a,b]$ para o melhor canal que assegure não haver sobreposição com outros.

Nos cenários dinâmicos, o intervalo de sorteio da distribuição Uniforme se altera de modo similar a uma lista circular. Com esse comportamento, a configuração $U[a_n,b_n]$ de cada canal n é substituída pela configuração $U[a_{n-1},b_{n-1}]$, seu canal vizinho à esquerda, a cada quantidade pré-definida de tentativas de transmissão. Essas alterações além de gerar dinamicidade no ambiente facilitam a análise dos resultados. Isso porque, é possível, por exemplo, identificar a qualquer instante o canal com a maior chance de sucesso.

Com essa descrição de comportamento, o cenário dinâmico A alterna o melhor canal a cada 20 mil tentativas de transmissão. A Figura 5.1 ilustra de modo esquemático

o comportamento de alternância entre as probabilidades no cenário dinâmico A.

Intervalos	Canais			
[0, 20000]	$U[a_0,b_0]$	$U[a_1,b_1]$	...	$U[a_n,b_n]$
[20000, 40000]	$U[a_n,b_n]$	$U[a_0,b_0]$	...	$U[a_{n-1},b_{n-1}]$
[40000, 60000]	$U[a_{n-1},b_{n-1}]$	$U[a_n,b_n]$	...	$U[a_{n-2},b_{n-2}]$
[60000, 80000]	$U[a_{n-2},b_{n-2}]$	$U[a_{n-1},b_{n-1}]$	...	$U[a_{n-3},b_{n-3}]$
[80000, 100000]	$U[a_{n-3},b_{n-3}]$	$U[a_{n-2},b_{n-2}]$	...	$U[a_{n-4},b_{n-4}]$

Figura 5.1: *Representação do comportamento de alternância entre o melhor canal para o cenário dinâmico A.*

No cenário dinâmico B, as mudanças de configuração ocorrem a cada 4 mil tentativas de transmissão, sendo portanto, mais dinâmico que o A. Logo, o cenário dinâmico B exige adaptação mais rápida dos mecanismos de DSS.

No cenário dinâmico C, assim como no B, a configuração dos canais também é alterada a cada 4 mil tentativas de transmissão, porém suas mudanças são diferentes dos outros dinâmicos apresentados. A cada alteração, é escolhido aleatoriamente um novo intervalo de sorteio $[a_n, b_n]$ para cada canal n e nessa escolha aleatória não é possível garantir que não haja sobreposição entre os canais. Assim, nesse cenário, é possível que o melhor canal se alterne dentro do intervalo de 4 mil tentativas de transmissões e, portanto, não há previsibilidade sobre seu comportamento.

5.2 Plataforma de simulação

O modelo apresentado na Seção 4.1 foi utilizado como base para a criação de um simulador customizado composto de duas partes, uma em Linguagem C e outra em Python. A parte em C utilizou a biblioteca GNU GSL (GNU *Scientific Library*) para gerar os diferentes cenários descritos na Seção 5.1 que representaram o comportamento dos canais sob condições variadas. A parte em Python simula as tentativas de transmissões com as diferentes abordagens avaliadas e mensura o desempenho dos algoritmos. O simulador passou por vários testes de verificação e validação para avaliar sua corretude e acurácia que serão detalhados nas seções a seguir.

A Figura 5.2 apresenta o diagrama de classes do simulador. Pelo diagrama nota-se que o Simulador é composto pelas classes Ambiente, Canal e SU. A classe SU por sua vez, é especializada em três tipos, de acordo com o algoritmo de DSS que o dispositivo irá empregar. Assim, as especializações são as seguintes: SuQl, SuEs e SuRand. A

especialização SuQl possui como atributos os parâmetros necessários a implementação de um dispositivo que use o *Q-Learning* para escolher o canal de transmissão: taxa de aprendizado (*alpha*), taxa de exploração (*epsilon*), valor máximo de Q (*qmax*), limiar para efetuar a troca entre canais (*beta*), tabela-Q (*qtable*) e tamanho da época (*td*).

A especialização SuEs implementa o ES^3 e o E^2S^3. Isso é possível porque existe o atributo *adaptativo* que ativa e desativa a característica de auto-adaptação que é o que realmente difere as versões dos algoritmos. Os outros atributos dizem respeito a parametrização do algoritmo (*sigma* e *td*), representação dos indivíduos e avaliação do seu desempenho (*lst_intervalos, lst_desempenho, lst_C* e *tam_gaussiana*) e ao processo de evolução (*num_geracoes*). A classe SuRand implementa o mecanismo Aleatório que não possui atributos.

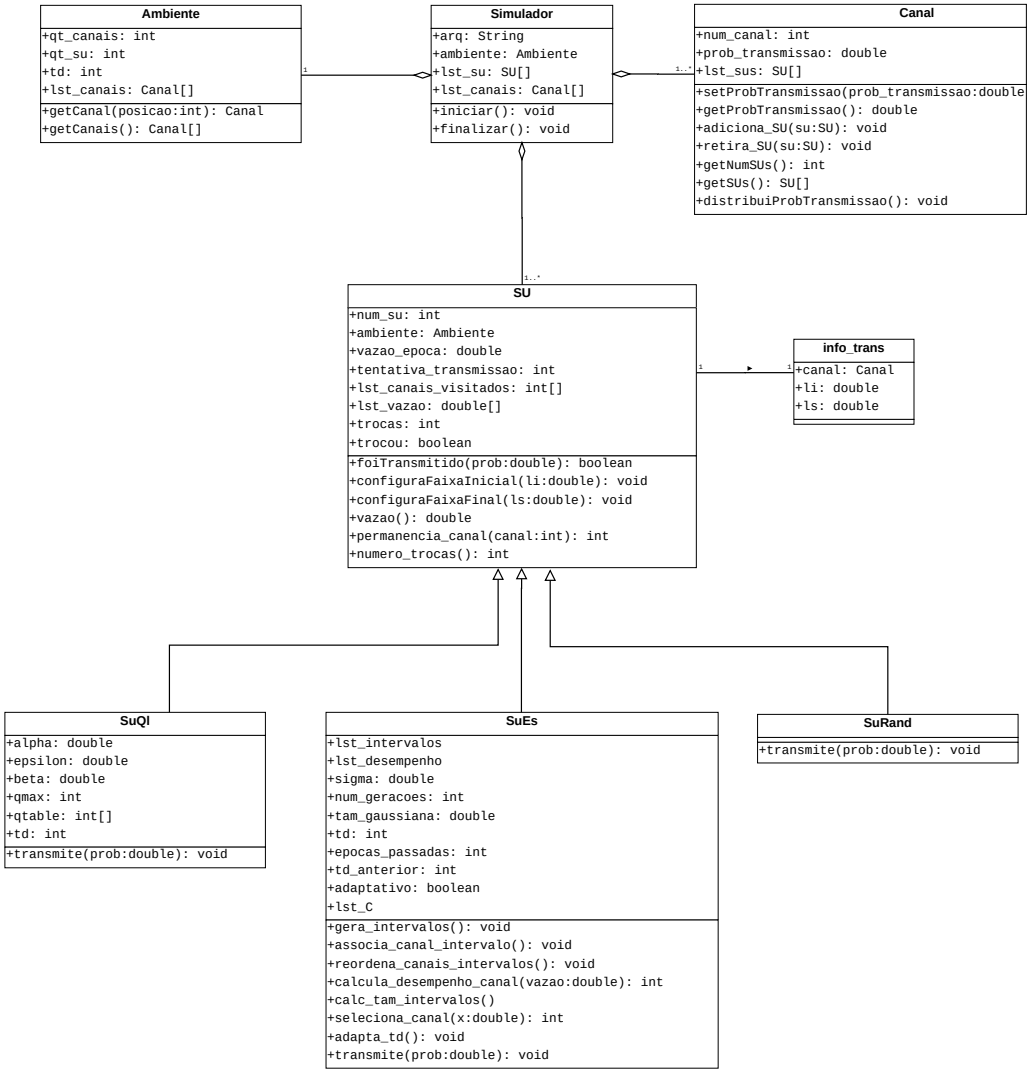


Figura 5.2: *Diagrama de classe da plataforma de simulação.*

5.3 *Q-Learning*

Nesta seção, são apresentados o algoritmo do *Q-Learning* utilizado neste trabalho e uma análise da influência de seus parâmetros.

5.3.1 Algoritmo

O *Q-Learning* avaliado neste trabalho foi baseado no algoritmo proposto em [48, 63] e está representado pelo Algoritmo 5.1. Os parâmetros de entrada são: t_D, α, ε, β e Q_{max}. Assim, seja $t \in T = \{1,2,...\}$ uma época e t_D a constante que representa a duração da época. Uma época corresponde a um determinado número de transmissões. Neste trabalho, $t_D = 30$, ou seja, cada época é composta por 30 tentativas de transmissão. Esse valor é o mesmo utilizado em [63]. Seja N_D a quantidade de transmissões bem sucedidas numa época. A função de recompensa é dada por $r : T \rightarrow \mathbb{R}$ tal que $r_t = N_D/t_D$ (linha 11), ou seja, a recompensa é dada pela vazão do canal escolhido. Uma ação a_t corresponde à escolha de um canal C_i para transmissão durante a época t. A primeira ação do algoritmo é aleatória (linha 3), mas as seguintes são baseadas nos critérios de exploração e aproveitamento.

A cada época, o algoritmo escolhe uma ação a_t e recebe uma recompensa $r_{t+1}(a_t)$ no tempo $t+1$. A política para tomada de uma ação é definida pela *tabela-Q*. O valor-Q inicial é igual a 1 para todas as ações (linha 2). O algoritmo atualiza o *valor-Q* de a_t (linha 12) conforme a Equação 5-1:

$$Q_{t+1}(a_t) \leftarrow (1-\alpha)Q_t(a_t) + \alpha r_{t+1}(a_t), \tag{5-1}$$

onde $0 \leq \alpha \leq 1$ é chamada de taxa de aprendizado. Maiores valores de α indicam maior peso para amostras recentes de recompensa, valoração do estado.

A escolha de uma ação é também influenciada por uma estratégia de exploração. A estratégia usada em [63] é a ε-*greedy* que seleciona uma ação de forma gulosa escolhendo aquela que possui o maior *valor-Q*, $\arg\max_{a \in A}(Q(a))$ (linha 18), e, eventualmente seleciona uma ação de maneira aleatória de acordo com ε (linha 16). Além disso, para melhorar a estabilidade do algoritmo, a ação não muda se a diferença entre os valores-Q da ação explorada anteriormente e a ação ótima atual for menor ou igual a um limiar β (linhas 20 a 23). Esse limiar permite que o mecanismo faça uma análise da possibilidade de mudança de canal, uma vez que isso acarreta custo de reconfiguração e sincronização, prejudicando a maximização da vazão. Portanto, esse parâmetro avalia a vantagem da adoção de uma ação numericamente melhor que a atual. Outro aspecto importante é o parâmetro Q_{max}, que limita o crescimento do valor-Q, pois seu crescimento indefinido

prejudicaria a convergência do algoritmo (linhas 13 e 14).

Algoritmo 5.1: *Q-Learning*

Entrada: t_D, α, ε, β, Q_{max}

1 $t \leftarrow 0$
2 $Q_t(a)_{a \in A} \leftarrow 1$
3 $a_t \leftarrow random(0,n)$
4 **enquanto** *houver transmissão* **faça**
5 $N_D \leftarrow 0$
6 **para** $j = 1 \rightarrow t_D$ **faça**
7 $sucesso \leftarrow transmitaEm(a_t)$
8 **se** $sucesso = True$ **então**
9 $N_D \leftarrow N_D + 1$
10 **fim**
11 $r_{t+1}(a_t) \leftarrow N_D/t_D$
12 $Q_{t+1}(a_t) \leftarrow (1-\alpha)Q_t(a_t) + \alpha r_{t+1}(a_t)$
13 **se** $(Q_{t+1}(a_t) > Q_{max})$ **então**
14 $Q_{t+1}(a_t) \leftarrow Q_{max}$
15 **se** $random(0,1) \leq \varepsilon$ **então**
16 $a_{t+1} \leftarrow random(0,n)$ `//exploração`
17 **senão**
18 $a_{temp} \leftarrow argmax_{a \in A}(Q(a))$ `//ação gulosa - aproveitamento`
19 **fim**
20 **se** $\left|Q_{t+1}(a_{temp}) - Q_{t+1}(a_t)\right| \leq \beta$ **então**
21 $a_{t+1} \leftarrow a_t$
22 **senão**
23 $a_{t+1} \leftarrow a_{temp}$
24 **fim**
25 **fim**
26 $a_{t+1} \leftarrow a_{temp}$
27 **fim**
28 $t \leftarrow t+1$
29 **fim**

5.3.2 Influência dos parâmetros

O algoritmo *Q-Learning* é capaz de obter resultados satisfatórios em vários cenários diferentes, no entanto, o seu desempenho sofre influência significativa de seus parâmetros de ajuste.

Conforme apresentado na Seção 5.3.1, os parâmetros para o *Q-Learning* são: taxa de aprendizado (α), taxa de exploração (ε), valor-Q máximo (Q_{max}), limiar para troca entre canais (β) e época de decisão (t_D). Nesta seção, são apresentados resultados que ilustram esse comportamento para os parâmetros Q_{max}, α e ε. Foram escolhidos esses três parâmetros para analisar, respectivamente, os comportamentos de convergência, de aprendizado e de exploração do algoritmo nos diferentes cenários.

As configurações para as simulações consistiram em fixar um dos parâmetros e variar o restante, com o algoritmo começando no pior canal do cenário avaliado e com $\beta = 1$ e $t_D = 30$. Assim, para a análise de convergência, Q_{max}, foram fixados os valores α em 0,5 e ε em 0,1. A escolha dos valores para os parâmetros fixos foi conservadora, isto é, baixa exploração e peso igual tanto para o histórico aprendido quanto para a recompensa atual (conforme Equação 5-1). Essa escolha visou interferir o mínimo possível nas características de aprendizado e exploração, independente do cenário.

Na análise de aprendizado, foram feitas duas análises, uma para Q_{max} igual a 5 e outra igual a 20. Nelas o parâmetro ε foi fixado em 0,1 enquanto α variava com passo de 0,05. Por último, na análise de exploração, também foram realizadas duas configurações. Uma com Q_{max} igual 5 e outra igual a 20. O parâmetro ε variava com passo de 0,05 e α foi fixado em 0,2. Esses valores, $\alpha = 0,2$ e $\varepsilon = 0,1$ e , foram adotados a partir de [63]. Os valores de $Q_{max} = 5$ e $Q_{max} = 20$ foram adotados com base em simulações realizadas durante a análise de convergência do *Q-Learning* que será apresentada na Seção 5.3.2.

Os resultados foram obtidos pela execução de 30 simulações de cada configuração e a média do percentual de permanência no melhor canal foi calculada com intervalo de confiança de 90%. Não são apresentados resultados para o cenário dinâmico C, pois conforme discutido na Seção 5.1, nesse cenário, o melhor canal pode variar a cada tentativa de transmissão.

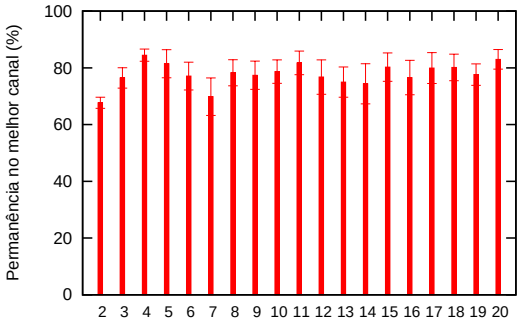
No geral, as análises apresentadas a seguir mostram como o ajuste de parâmetros do *Q-Learning* pode se tornar uma desvantagem. Duas dificuldades são mais notáveis, uma é encontrar o valor adequado de um determinado parâmetro para um cenário específico, pois esse processo envolve aspectos de modelagem e/ou de mensuração de resultados. A segunda dificuldade é lidar com ambientes que demandem novos ajustes com alta regularidade, reduzindo severamente a característica autonômica do algoritmo.

Análise de convergência

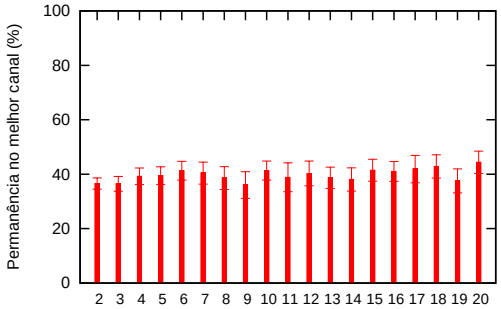
Em relação a convergência, as Figuras 5.3(a), 5.3(b) e 5.3(c) apresentam os resultados obtidos através da variação do valor-Q limite, Q_{max}, para os cenários estático, dinâmico A e dinâmico B, respectivamente. Em todas elas, o percentual de permanência no melhor canal oscila bastante e isso pode ser observado pela margem de erro para um mesmo valor de Q_{max}.

No cenário estático, Figura 5.3(a), o melhor desempenho foi obtido com Q_{max} igual a 4. No caso do cenário dinâmico A, Figura 5.3(b), o melhor desempenho foi alcançado com Q_{max} igual a 20. Em relação ao cenário dinâmico B, Figura 5.3(c), o melhor desempenho foi alcançado com Q_{max} igual a 18.

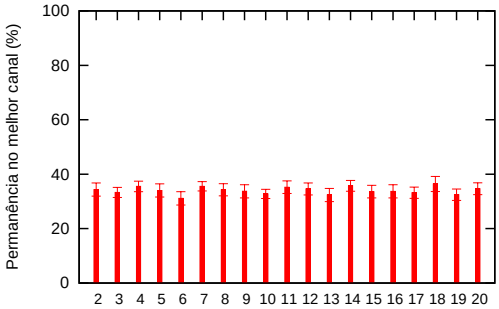
Observa-se que Q_{max} é importante para que o valor-Q não cresça indefinidamente, pois de acordo com o tipo de cenário, esse crescimento pode prejudicar a escolha entre os canais durante uma ação de aproveitamento. Isso se deve ao limiar β. Por exemplo, se o valor-Q do segundo melhor canal é muito alto em relação ao do melhor canal, o limiar β pode não ser alcançado e, portanto, outros canais, mesmo que melhores, podem não ser escolhidos. Essa é uma situação prejudicial tanto no cenário estático quanto no dinâmico. No estático isso significa ignorar oportunidades melhores e nos dinâmicos, não reagir tão rapidamente quanto às mudanças que estão acontecendo no ambiente. Do comportamento de Q_{max} depreende-se que quanto mais dinâmico o cenário, menor é o impecto de Q_{max}.



(a) *Cenário estático.*

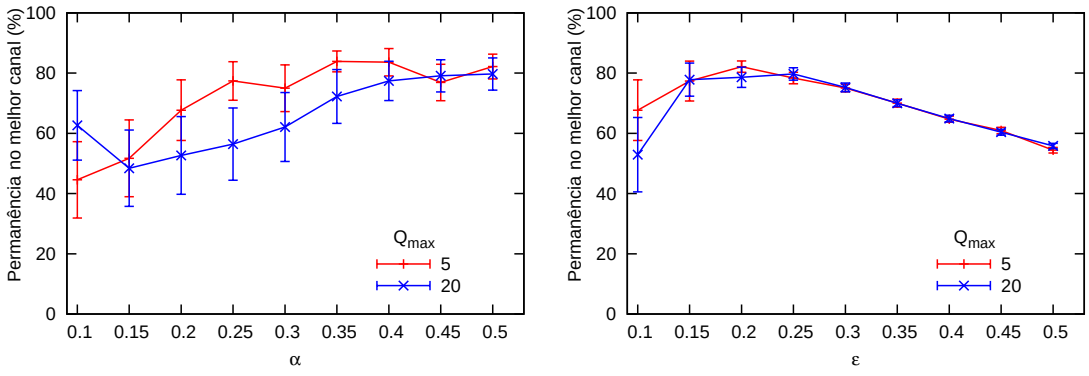


(b) *Cenário dinâmico A.*



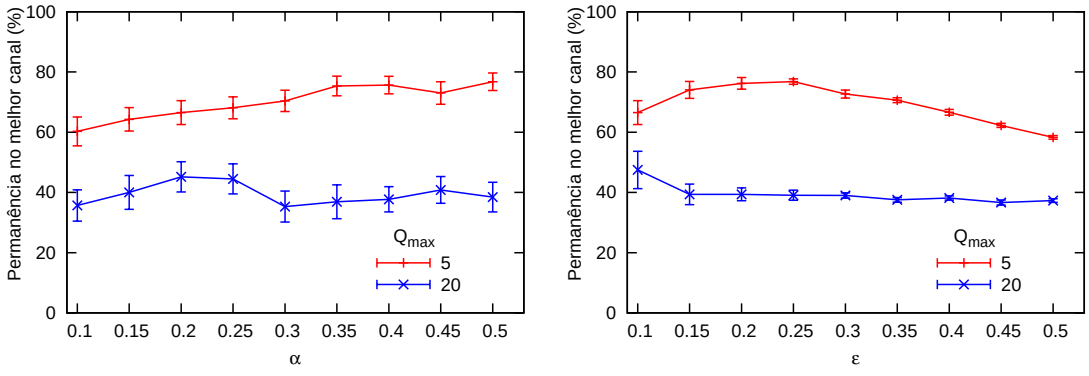
(c) *Cenário dinâmico B.*

Figura 5.3: *Impacto do parâmetro Q_{max}.*



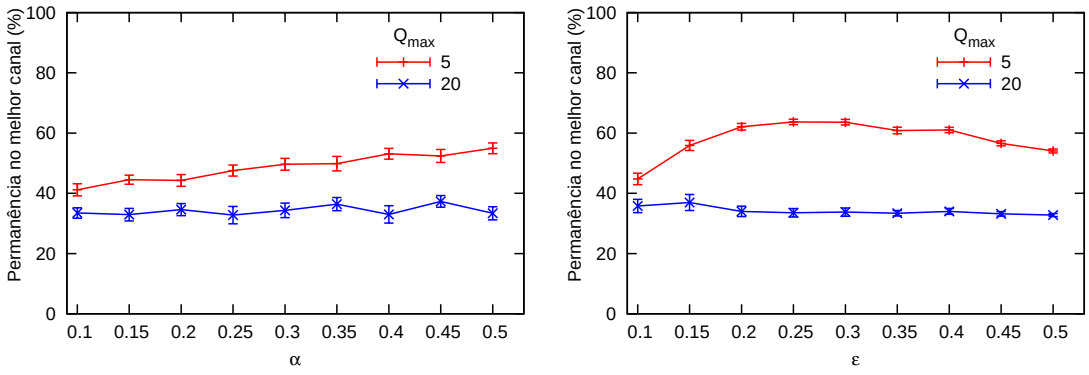
(a) α. (b) ε.

Figura 5.4: *Impacto dos parâmetros no cenário estático.*



(a) α. (b) ε.

Figura 5.5: *Impacto dos parâmetros no cenário dinâmico A.*



(a) α. (b) ε.

Figura 5.6: *Impacto dos parâmetros no cenário dinâmico B.*

Análise do aprendizado

Em relação ao aprendizado, as Figuras 5.4(a), 5.5(a) e 5.6(a) apresentam os resultados obtidos através da variação de α para os cenários estático, dinâmico A e dinâmico B, respectivamente. No geral, na maioria dos valores de α alcançou maior percentual quando $Q_{max} = 5$.

No cenário estático, ilustrado na Figura 5.4(a), o melhor desempenho foi obtido com α superior a 0,3 para $Q_{max} = 5$ e superior a 0,35 para $Q_{max} = 20$. Esses são valores muito próximos, mesmo com uma diferença muito grande entre Q_{max}.

No cenário dinâmico A, ilustrado na Figura 5.5(a), é possível observar que α possui melhor desempenho com valores superiores a 0,3 para $Q_{max} = 5$ e inferiores a 0,3 para $Q_{max} = 20$. A alteração do valor Q_{max} produziu uma inversão na tendência de comportamento da curva, mostrando que maior valor de Q_{max} requereu menor valor de α e menor valor de Q_{max} requereu maior valor de α nesse cenário. Esse é um dos reflexos da avaliação de convergência apresentada anteriormente e comprova o impacto do parâmetro Q_{max} no desempenho geral do algoritmo. O valor observado é o mesmo para o cenário estático, isso pode ser atribuído ao fato de que o melhor canal é alterado apenas cinco vezes durante todo o tempo de simulação.

No cenário dinâmico B, ilustrado na Figura 5.6(a), α possui melhor desempenho com valores superiores a 0,35 para $Q_{max} = 5$ e superiores a 0,4 para $Q_{max} = 20$. Esse valor é maior que nos cenários anteriores. Isso é coerente com o esperado, pois nesse cenário o melhor canal se alterna 25 vezes, ou seja, a dinacimicidade é alta.

Diante disso, a taxa de aprendizado mostrou-se mais tolerante em relação ao aumento da dinamicidade do ambiente, visto que só apresentou aumento quando o nível de dinamicidade foi elevado. Outro fator importante é o fato do primeiro canal escolhido pelo *Q-Learning* não ser o melhor e, portanto, ele não deve colocar um peso muito alto no histórico das transmissões. Essa estratégia também é justificada pela diferença entre o melhor canal e o segundo melhor em todos os cenários não ser muito alta.

Análise da exploração

A respeito da exploração, as Figuras 5.4(b), 5.5(b) e 5.6(b) apresentam resultados obtidos através da variação de ε para os cenários estático, dinâmico A e dinâmico B, respectivamente.

Na Figura 5.4(b), o melhor desempenho foi obtido com o valor de ε em 0,2 para $Q_{max} = 5$ e 0,25 para $Q_{max} = 20$. Esses valores se mostram adequados, pois em um cenário estático, onde o melhor canal é conhecido, a ação de exploração tem sua importância até que o melhor canal seja encontrado. Depois, somente a ação de aproveitamento fará com que o dispositivo permaneça naquele canal considerado melhor. Por isso, com valores de ε

superiores ao destacado, o percentual de permanência no melhor canal sofre diminuição. Outro ponto de destaque, é que as curvas para $Q_{max} = 5$ e $Q_{max} = 20$ são praticamente coincidentes. Isso demonstra que nesse cenário o parâmetro Q_{max} quase não influenciou o comportamento de exploração do algoritmo, comprovado pela diferença de apenas 0,05 entre os valores de ε destacados.

Na Figura 5.5(b), é possível observar que ε possui um máximo em 0,25 para $Q_{max} = 5$ e 0,1 para $Q_{max} = 20$. Como nesse cenário o melhor canal muda 5 vezes, é necessário que a estratégia de exploração seja adequada para contribuir com o processo de aprendizado.

Na Figura 5.6(b), o percentual máximo é alcançado com ε em 0,3 para $Q_{max} = 5$ e 0,15 para $Q_{max} = 20$. Aumento de 0.05 em relação ao dinâmico A e de 0.1 em relação ao cenário estático. O nível de dinamicidade desse cenário exigiu uma taxa de exploração maior para que alcançasse mais sucesso no processo de aprendizado.

A necessidade de exploração, de acordo com o aumento no valor de Q_{max}, foi minimizada nos cenários dinâmicos. A existência de um ótimo era esperado para a taxa de exploração. No entanto, esse valor é diferente de acordo com o nível de dinamicidade do cenário. Em geral, ambientes mais dinâmicos exigem maiores valores de ε, enquanto ambientes menos dinâmicos toleram valores menores. Isso é refletido na suavização que ocorre na curva de exploração.

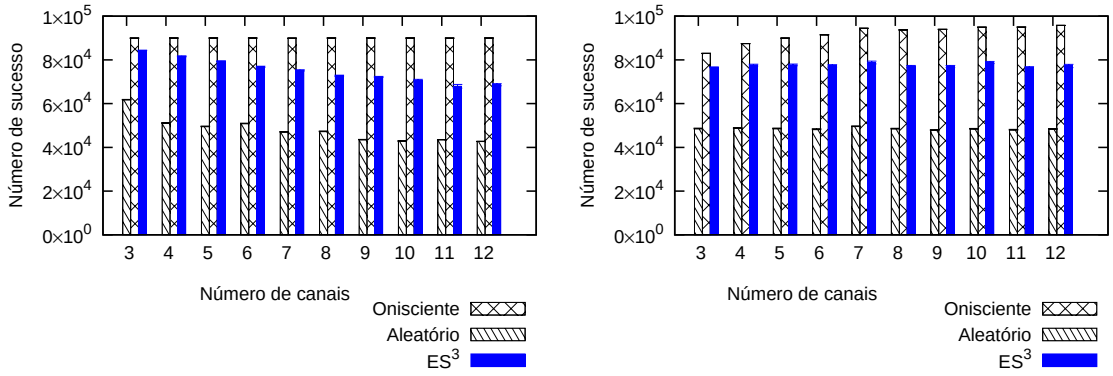
5.4 Comparação entre os mecanismos

Nesta seção, serão avaliados os algoritmos e analisados de modo comparativo os resultados obtidos. Para isso, as métricas avaliadas são o número de sucessos e os canais visitados.

Na Tabela 5.1, são apresentados os parâmetros do *Q-Learning*, do ES^3 e do E^2S^3, utilizados nos experimentos. Foi escolhida a configuração do *Q-Learning* presente em [63].

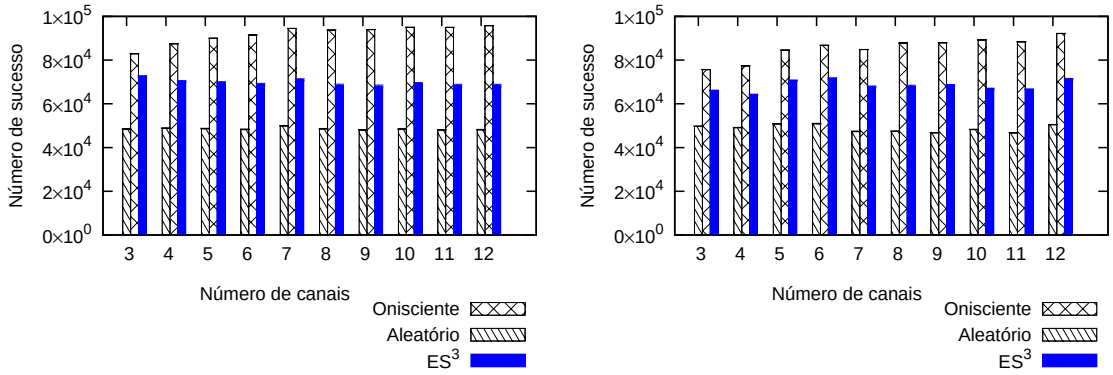
Tabela 5.1: *Parâmetros de simulação.*

Algoritmos	$t_D inicial$	α	ε	β	Q_{max}	σ	k
Q-Learning	30	0,2	0,2	1	20	-	-
ES^3	30	-	-	-	-	1	33
E^2S^3	30	-	-	-	-	-	-



(a) *Cenário estático.* (b) *Cenário dinâmico A.*

Figura 5.7: *Desempenho dos algoritmos em cenários com menor dinamicidade para um par SU.*



(a) *Cenário dinâmico B.* (b) *Cenário dinâmico C.*

Figura 5.8: *Desempenho dos algoritmos em cenários com maior dinamicidade para um par SU.*

5.4.1 Avaliação da corretude dos algoritmos

Para a verificação e validação do simulador e da corretude dos algoritmos foi desenvolvido um mecanismo de referência chamado Onisciente. É importante destacar que esse mecanismo não pode ser implementado no mundo real, pois ele realiza simultaneamente duas ações distintas: verificar a qualidade de todos os canais disponíveis e escolher o canal com a maior chance de sucesso.

As Figuras 5.7(a), 5.7(b), 5.8(a) e 5.8(b) mostram o número de sucessos com um par SU para os algoritmos Onisciente, Aleatório e ES^3 nos cenários estático, dinâmico A, dinâmico B e dinâmico C, respectivamente. No cenário estático, o intervalo de sorteio $[a_{mc}, b_{mc}]$ do melhor canal, mc, foi fixado e, portanto, o desempenho do Onisciente foi idêntico para qualquer número de canais. No cenário dinâmico A, o intervalo de sorteio do melhor canal teve seu limite inferior a_{mc} aumentado proporcionalmente ao número de canais, enquanto $b_{mc} = 1$. Por essa razão, o desempenho do Onisciente e dos demais mecanismos aumenta conforme cresce o número de canais no cenário dinâmico A.

Apesar das mudanças no cenário dinâmico B serem mais frequentes que no dinâmico A, a configuração do melhor canal é idêntica. Assim, no cenário dinâmico B, o desempenho do mecanismo Onisciente é idêntico ao obtido no cenário dinâmico A. No cenário dinâmico C, o comportamento do Onisciente não é previsível como nos cenários anteriores, mas há uma pequena tendência de aumento no desempenho proporcional ao número de canais. Isso ocorre porque um número maior de canais implica em uma maior chance de algum canal estar melhor, mesmo que durante uma única tentativa de transmissão.

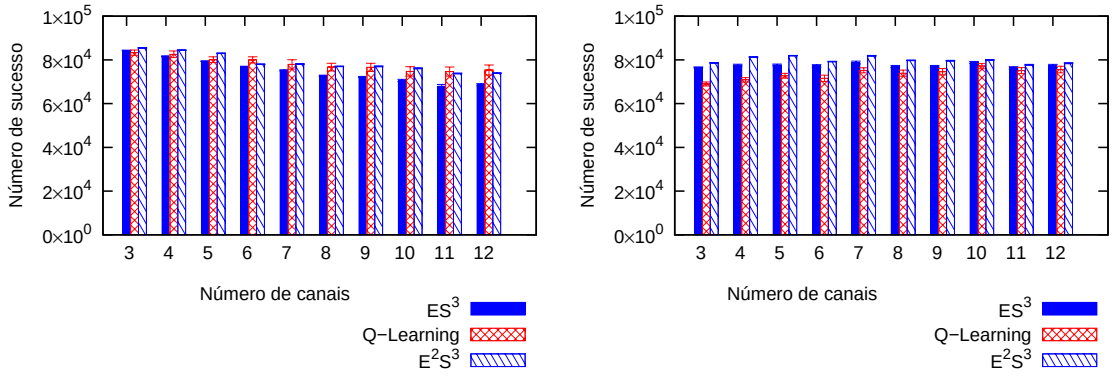
O ES^3 também foi comparado ao método de seleção aleatório com a finalidade de comprovar que a aplicação de estratégia evolutiva ao mecanismo de seleção dinâmica de espectro poderia proporcionar maior ganho no uso do espectro. Assim, conforme as Figuras 5.7(a), 5.7(b), 5.8(a) e 5.8(b), nos cenários dinâmicos o método Aleatório produz resultados próximos de 50 mil, sem muita variação dentro de um mesmo cenário, ainda que o número de canais aumente. A maior variação ocorreu no cenário estático, Figuras 5.7(a), na qual com uma quantidade pequena de canais o número de sucessos alcança 60 mil, que diminui de acordo com o acréscimo da quantidade de canais.

Em relação ao ES^3, no cenário estático, Figura 5.7(a), com poucos canais o número de sucessos alcançou 80 mil e com o aumento gradativo de canais esse valor diminui para 70 mil. No cenário dinâmico A, Figura 5.7(b), o número de sucessos mesmo com o aumento do número de canais gira em torno de 80 mil com pouca variação mesmo com o aumento da quantidade de canais. Para o dinâmico B, Figura 5.8(a), o número de sucessos com o aumento do número de canais, com comportamento parecido ao dinâmico A fica em torno de 75 mil. O cenário dinâmico C, Figura 5.8(b), o número de sucessos varia no patamar de 70 mil, porém com maior oscilação. Esse comportamento é devido a dinamicidade maior que o dinâmico B.

Dessa avaliação foi observado que o mecanismo de seleção baseado em estratégia evolutiva permite bom aproveitamento do espectro em relação a um mecanismo de escolha simplesmente aleatória.

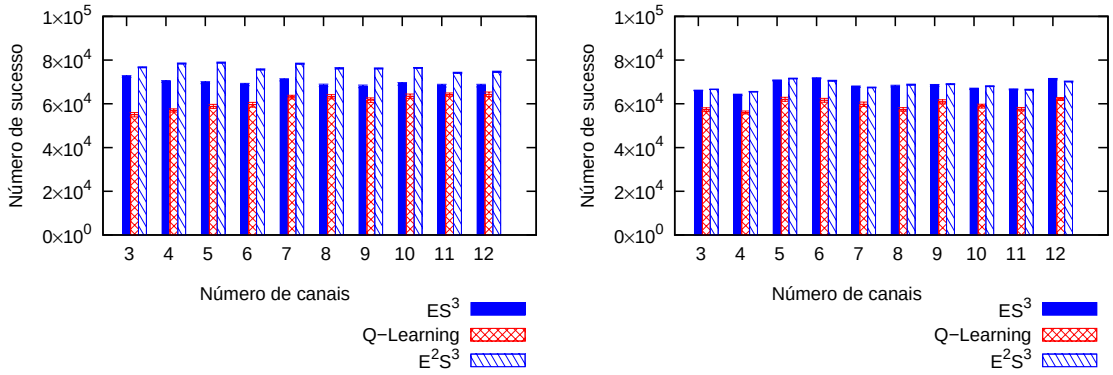
5.4.2 Análise de desempenho para um par SU

Nesta seção, são feitas as considerações sobre os resultados obtidos com os mecanismos *Q-Learning*, ES^3 e E^{2}S^3. Nas Figuras 5.9(a) e 5.9(b), são exibidos os resultados dos mecanismos em cenários nos quais a qualidade média do canal se altera pouco (dinâmico A) ou nada (estático). Essas figuras mostram a média do número de tentativas de sucesso em função do número de canais para 30 execuções. Os valores podem ser conferidos nas Tabelas 5.2 e 5.3. O número de sucessos do ES^3 é muito similar à do *Q-Learning* no cenário estático. Esse resultado era esperado, porque o



(a) *Cenário estático.* (b) *Cenário dinâmico A.*

Figura 5.9: *Desempenho dos algoritmos em cenários com menor dinamicidade para um par SU.*



(a) *Cenário dinâmico B.* (b) *Cenário dinâmico C.*

Figura 5.10: *Desempenho dos algoritmos em cenários com maior dinamicidade para um par SU.*

ambiente proporciona ao *Q-Learning* um longo período para aprendizado, permitindo que ele identifique o melhor canal e evite transmissões nos demais canais. Tanto *Q-Learning* quanto ES^3 e o E^2S^3 sofrem com o aumento do número de canais, uma vez que aumenta o espaço de busca pelo melhor canal. No cenário dinâmico A, o E^2S^3 exibe desempenho médio superior ao do *Q-Learning* e ao ES^3, em alguns casos a diferença alcança até 10 mil.

Os resultados obtidos nos cenários com variação mais frequente na qualidade média do canal são apresentados nas Figuras 5.10(a) e 5.10(b). Os valores podem ser conferidos nas Tabelas 5.4 e 5.5. Apesar das mudanças no cenário dinâmico B serem mais frequentes que no dinâmico A, a configuração do melhor canal é idêntica. Houve redução no número médio de sucesso do *Q-Learning*, do ES^3 e do E^2S^3, pois as alterações mais frequentes do cenário dinâmico B levam os algoritmos a permanecerem em canais diferentes do melhor. Apesar disso, a característica de auto-adaptação aplicada ao E^2S^3 permite que nesse cenário o número de sucessos seja superior, permanecendo próximo

Tabela 5.2: *Número de sucessos em cada canal para cada algoritmo com um par de SU no cenário estático*

Quantidade de Canais	ES^3	*Q-Learning*	E^2S^3
3	84.324,30	82.636,23	85.470,77
4	81.633,27	81.127,57	84.476,40
5	79.431,03	80.352,63	83.070,17
6	76.845,73	79.475,80	78.052,33
7	75.191,87	78.556,30	78.091,27
8	72.736,93	77.931,70	77.040,60
9	72.209,80	78.048,70	77.064,47
10	70.599,17	77.471,80	76.156,60
11	67.936,00	76.343,70	73.825,33
12	68.661,80	76.216,73	73.959,40

Tabela 5.3: *Número de sucessos em cada canal para cada algoritmo com um par de SU no cenário dinâmico A*

Quantidade de Canais	ES^3	*Q-Learning*	E^2S^3
3	76.605,83	68.972,73	78.629,07
4	77.583,03	69.679,30	81.312,20
5	77.715,83	71.143,73	81.955,40
6	77.239,73	71.228,90	79.232,90
7	78.812,60	73.167,17	81.881,23
8	76.966,53	72.271,57	79.775,37
9	76.999,00	72.110,87	79.609,90
10	78.835,90	75.530,83	79.954,07
11	76.331,40	73.233,13	77.749,67
12	77.251,03	73.761,83	78.494,43

dos 80 mil.

No cenário dinâmico C, Figura 5.10(b), por exemplo, o ES^3 alcança uma diferença superior a 14 mil com 3 canais enquanto o E^2S^3 é cerca de 15 mil.

Novamente, o ES^3 teve desempenho superior ao do *Q-Learning* independente da quantidade de canais. A vantagem do ES^3 sobre o *Q-Learning* esteve sempre em torno de 10 mil e do E^2S^3 em torno de 20 mil, mesmo quando havia muitos canais disponíveis.

5.4.3 Avaliação de desempenho para múltiplos pares SU

Nesta seção, são feitas as considerações sobre os resultados obtidos com múltiplos pares SU com os mecanismos *Q-Learning*, ES^3 e E^2S^3. As Figuras 5.11, 5.12, 5.13 e 5.14 resumem os resultados de todos os mecanismos para múltiplos pares SUs em cada um dos cenários. Essas figuras mostram a média do número de tentativas de sucesso em

Tabela 5.4: *Número de sucessos em cada canal para cada algoritmo com um par de SU no cenário dinâmico B*

Quantidade de Canais	ES^3	*Q-Learning*	E^2S^3
3	72.484,10	55.859,30	76.705,70
4	70.207,97	58.686,73	78.440,90
5	69.648,50	58.641,83	78.832,97
6	68.817,07	59.804,80	75.724,00
7	70.957,97	63.918,17	78.365,30
8	68.413,20	61.593,97	76.225,37
9	67.987,27	62.616,53	76.165,27
10	69.094,80	62.888,03	76.447,60
11	68.295,70	62.873,80	74.188,10
12	68.310,97	64.087,37	74.652,30

Tabela 5.5: *Número de sucessos em cada canal para cada algoritmo com um par de SU no cenário dinâmico C*

Quantidade de Canais	ES^3	*Q-Learning*	E^2S^3
3	66.097,43	57.607,80	66.614,33
4	64.083,77	57.012,27	65.515,13
5	70.707,67	62.084,27	71.556,27
6	71.703,97	62.340,30	70.551,03
7	67.825,23	60.462,80	67.476,47
8	68.103,27	59.337,23	68.756,97
9	68.452,23	62.138,67	68.974,97
10	66.868,97	59.432,03	68.050,83
11	66.596,90	59.069,67	66.478,60
12	71.292,57	63.867,47	70.209,23

função do número de canais para 30 execuções. As colunas nas figuras possuem 4 segmentos. Cada um deles apresenta a média para uma quantidade específica de pares SU: 3, 6, 9 ou 12.

O nível de contenção entre os pares SU possui importante influência nos mecanismos. Percebe-se que no cenário estático o E^2S^3 é superado pelo *Q-Learning* sendo a diferença maior com o aumento da concorrência. Porém, nos cenários dinâmico A, B e C, o E^2S^3 possui desempenho superior ao do *Q-Learning* que se torna maior ainda quando a concorrência aumenta.

De modo geral, o *Q-Learning* possui melhor desempenho no cenário estático, enquanto o E^2S^3 supera o *Q-Learning* nos demais cenários. Esse comportamento é resultado do projeto dos algoritmos.

O *Q-Learning* experimenta os canais em uma velocidade menor porque possui uma época fixa e somente após preenchida a tabela-Q que uma escolha consciente

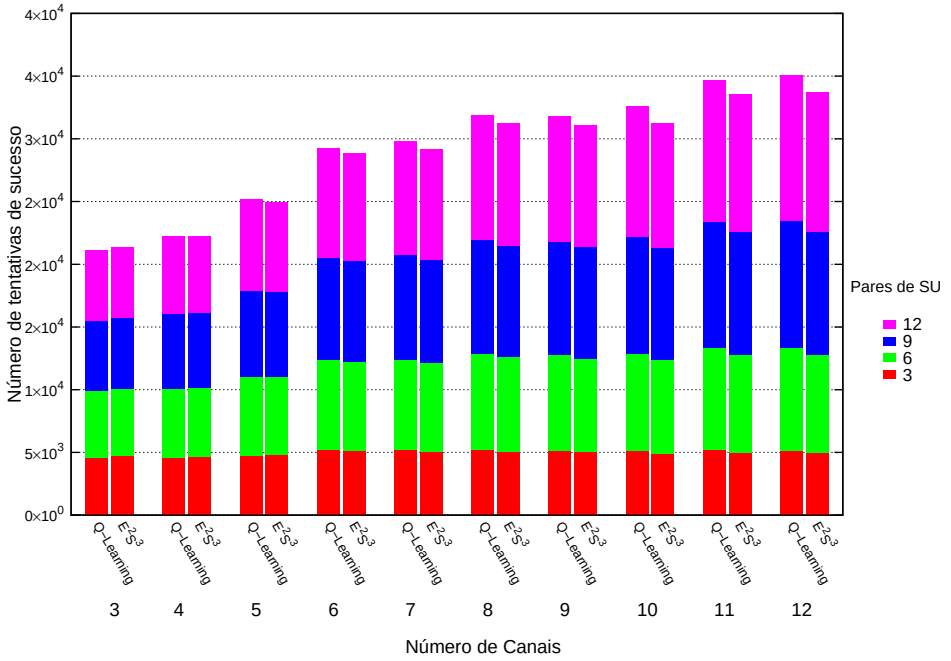


Figura 5.11: *Comparação de desempenho no cenário estático.*

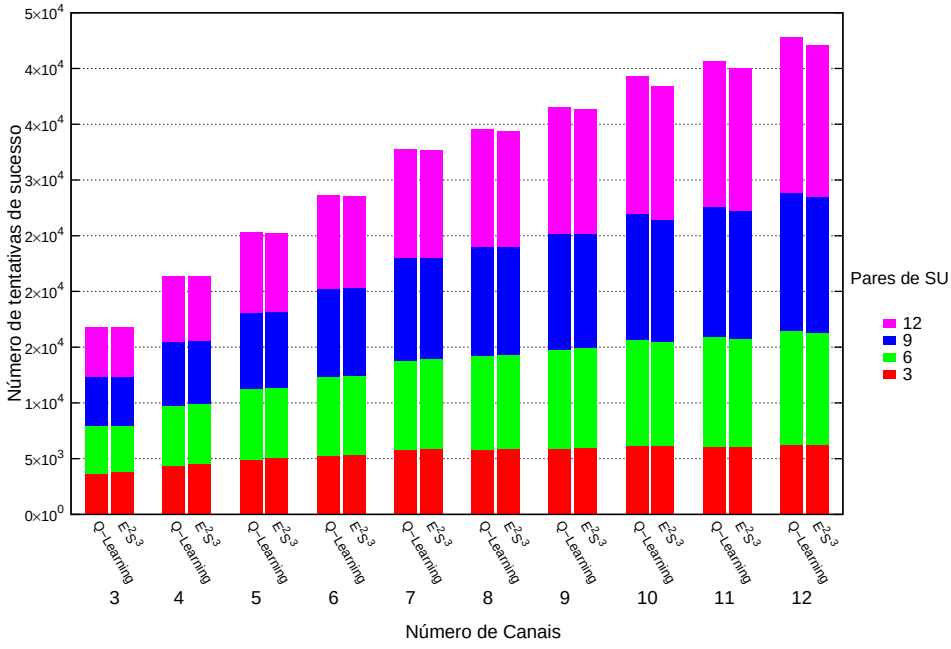


Figura 5.12: *Comparação de desempenho no cenário dinâmico A.*

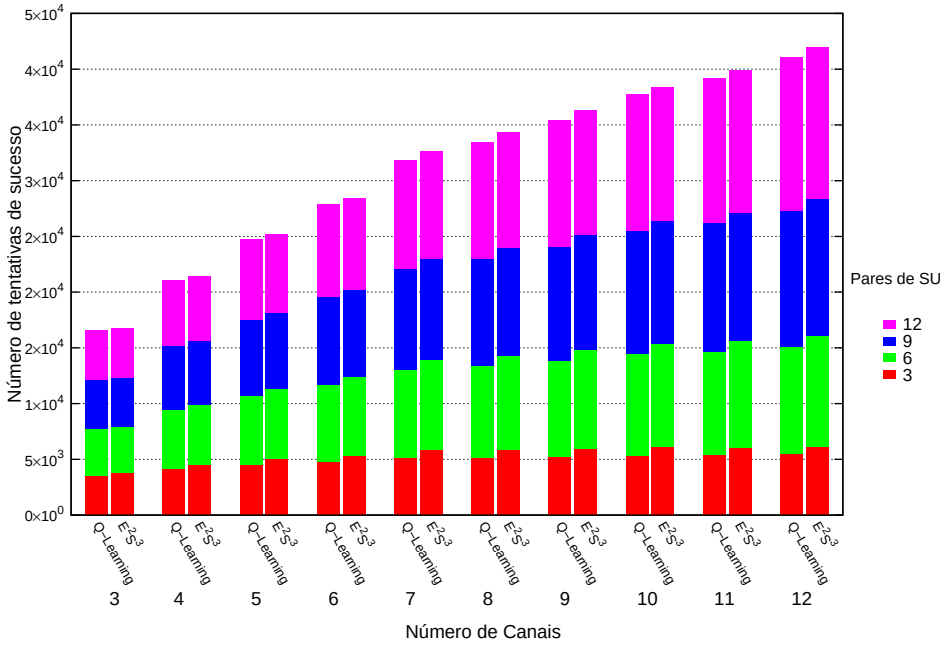


Figura 5.13: *Comparação de desempenho no cenário dinâmico B.*

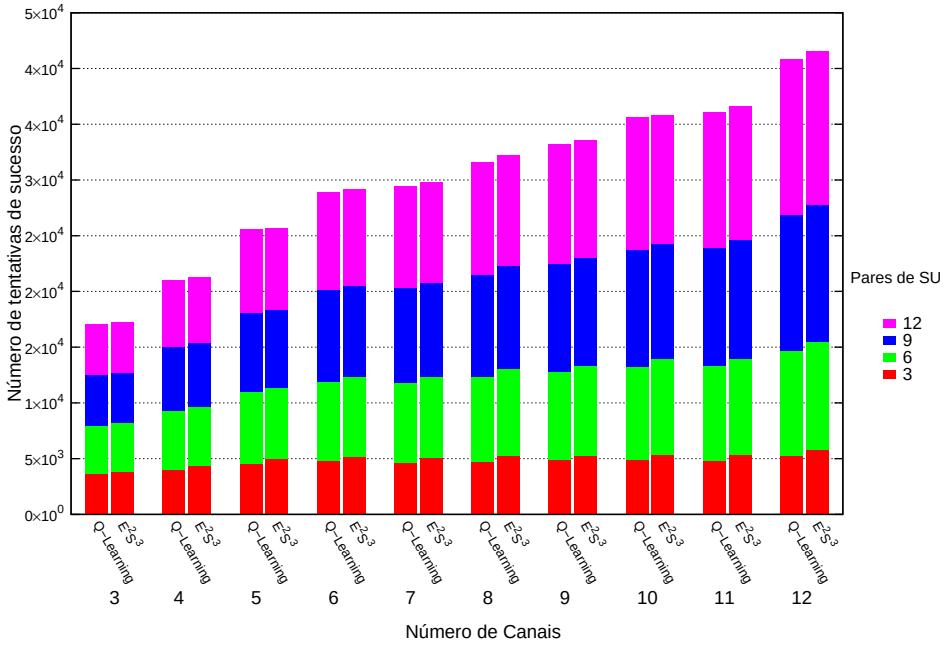


Figura 5.14: *Comparação de desempenho no cenário dinâmico C.*

Tabela 5.6: *Soma do número de sucessos por cenário.*

	Q-Learning	ES^3	E^2S^3
Estático	10.595.592,40	10.119.933,47	10.380.759,33
Dinâmico A	11.356.554,43	10.877.717,47	11.340.962,30
Dinâmico B	10.925.750,03	10.683.955,93	11.290.241,33
Dinâmico C	10.589.195,37	10.443.570,10	10.794.413,87

pode ser realmente feita. Em cenários onde as oportunidades de transmissão variam gradativamente, a tabela-Q oferece em termos numéricos uma boa representação da capacidade disponível dos canais. Em cenários onde as oportunidades de transmissão são mais dinâmicas, a taxa de exploração ajuda na busca do ainda desconhecido melhor canal.

O E^2S^3 experimenta os canais a uma taxa adaptativa que é alterada de acordo com o número de sucessos das tentativas de transmissão. Essa abordagem é bastante eficaz em cenários dinâmicos. Além disso, E^2S^3 retém um histórico compacto de informações sobre os canais e, por essa razão, o pior canal ainda possui chances não negligenciáveis de ser escolhido. Em cenários estáticos, essa abordagem faz com que o mecanismo saia do melhor canal com uma regularidade maior que a necessária. O projeto do E^2S^3 foi feito para que o algoritmo fosse mais adequado para ambientes dinâmicos. Essa decisão foi baseada no fato de que, no mundo real, cenários estáticos são incomuns [64].

A Tabela 5.6 mostra o somatório do número de sucessos obtidos em todos os experimentos separados por cenário, comprovando que o E^2S^3 alcançou o objetivo de seu projeto: ser mais adequado para ambientes dinâmicos. Esses valores indicam que para os cenários de menor dinamicidade o *Q-Learning* possui melhor desempenho, enquanto para os cenários de maior dinamicidade, o E^2S^3 sobressai. O *Q-Learning* apresenta diferença superior ao E^2S^3 de 2% no cenário estático e no cenário dinâmico o E^2S^3 supera o *Q-Learning* em 1,6%

A Tabela 5.7 mostra o somatório do número de sucessos para os pares 1, 3, 6, 9 e 12 em todos os cenários e em todos os canais. Observa-se que o E^2S^3 apresenta bons resultados para 1, 3 e 6 pares e perde desempenho para 9 e 12 pares, porém a diferença fica em torno de 1% para 9 pares e de 3% para 12 pares.

Esses valores demonstram que o projeto do E^2S^3 está adequado a ambientes de menor concorrência, mas necessita de melhoria para atuar em ambientes de maior competição.

5.4.4 Verificação do tempo de reação dos algoritmos

Para ilustrar porque o ES^3 e o E^2S^3 apresentam melhor desempenho que o *Q-Learning*, sobretudo quando a dinamicidade do ambiente aumenta, um experimento

Tabela 5.7: *Soma do número de sucessos por pares.*

Pares	*Q-Learning*	ES^3	E^2S^3
1	2.723.584,27	2.899.860,33	3.035.732,90
3	6.672.311,83	6.498.403,73	7.002.262,93
6	9.895.281,27	9.435.612,17	9.935.142,37
9	11.577.234,43	11.116.559,03	11.459.492,83
12	12.598.680,43	12.174.741,70	12.373.745,80

do cenário dinâmico B com 3 canais disponíveis foi escolhido para uma análise mais detalhada. A Figura 5.15 mostra os canais visitados pelo par SU em cada mecanismo de DSS avaliado. O comportamento do mecanismo Onisciente mostra a estratégia ótima para a escolha dos canais. Ao comparar o perfil do comportamento dos mecanismos nota-se que o ES^3 e do E^2S^3 são os que mais se aproximam do perfil de referência produzido pelo mecanismo Onisciente. No geral, é possível observar que o E^2S^3 e ES^3 conseguem identificar e reagir mais rapidamente às mudanças que o *Q-Learning*. Nesse cenário, o aprendizado do *Q-Learning* tende a mantê-lo em um canal durante período excessivamente longo a ponto das mudanças no ambiente não gerarem reação.

A característica de adaptação adotada no E^2S^3 permite que a convergência para o melhor canal ocorra de forma rápida. Esse comportamento pode ser observado na Figura 5.15 durante as primeiras 30 mil tentativas de transmissão. Nesse intervalo, o E^2S^3 acerta mais que o ES^3 que, por sua vez, acerta mais que o *Q-Learning*. Após 30 mil tentativas de transmissão, o E^2S^3 tem a sua reação atrasada porque permanece por muitos instantes de transmissão naquele canal que estava melhor no intervalo anterior. Embora visite o melhor canal para o intervalo, mas não o suficiente para permanecer. Nesse mesmo intervalo analisado, os outros mecanismos nem visitam o melhor canal. Esse fato acontece por exemplo, entre os instantes 30 mil e 50 mil. Essa estratégia do E^2S^3 é a que mais se aproxima da estratégia ótima e consequentemente, produz resultados melhores.

5.4.5 Avaliação da justiça

No contexto de múltipos SUs, a concorrência entre os pares poderia gerar diferença no aproveitamento do espectro, isto é, poderia haver benefício de alguns pares em detrimento de outros. Isso é um comportamento comum quando existe uma grande concorrência e pouco recurso disponível, principalmente quando o método é guloso. Nesse sentido, foi realizada a avaliação da justiça dos algoritmos através da medição do índice de justiça de Raj Jain [31]. Esse índice é dado pela seguinte Equação 5-2.

$$\theta = \frac{(\sum_i \rho_i)^2}{n * \sum_i (\rho_i)^2}, \tag{5-2}$$

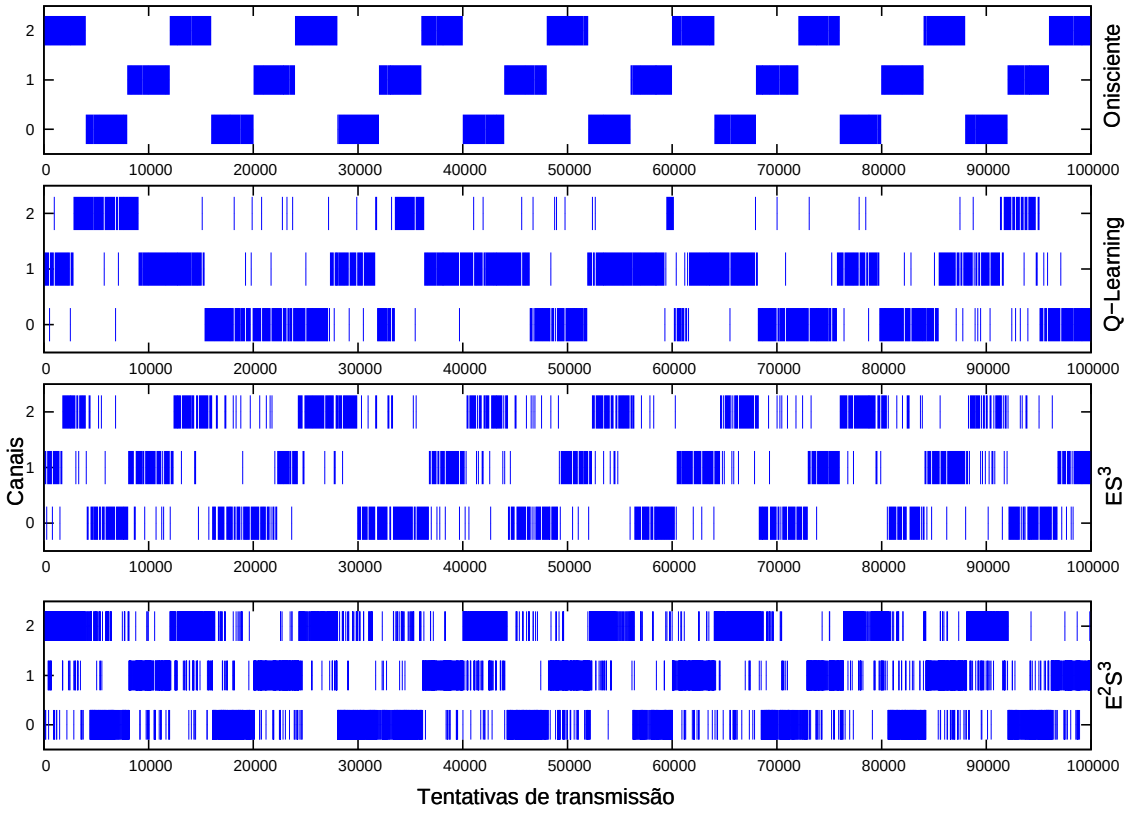


Figura 5.15: *Trocas de canais no cenário dinâmico B com 3 canais.*

onde θ representa o índice de justiça, ρ o número de sucessos para cada par SU identificado por i e n a quantidade de pares SU concorrentes. Todas as execuções tiveram o índice calculado e os resultados se aproximavam de 1 diferindo-se apenas na quarta ou quinta casa decimal. Assim, o algoritmo projetado mostra-se justo mesmo em cenários bastante concorridos.

Para analisar o resultado obtido, duas situações devem ser levadas em consideração: pares SU distribuídos entre os canais e pares SU em um mesmo canal. Na primeira situação, a tendência de comportamento é que cada par SU escolha, na maioria das vezes, senão o melhor canal, aqueles que são melhores, uma vez que a escolha é baseada no desempenho obtido em épocas anteriores. Assim, a chance de sucesso na transmissão é alta e todos eles podem conseguir transmitir, havendo uma equivalência de desempenho obtido entre os pares. Mesmo que os pares SU estejam distribuídos entre canais ruins, o algoritmo foi projetado para que reagisse rapidamente e percebesse que não fez uma boa escolha. Essa reação rápida permite que o desempenho de um par SU não seja tão ruim quando comparado a outro par SU.

Na situação em que os pares SU que tenham escolhido o mesmo canal, a priorização na transmissão é o ponto-chave para promover a justiça entre os pares alocados. Dada a probabilidade do canal, esse valor é fatiado igualmente e distribuído

entre os pares SU que estão na fila alocados para aquele canal. A probabilidade para transmissão sorteada é então comparada entre as fatias atribuídas a cada par SU. Aquele par SU cuja fatia corresponda à probabilidade de transmissão sorteada terá direito de transmitir naquele instante.

Ambas situações contribuem para que todos os pares SU consigam aproveitar de modo equivalente as chances de transmitir e isso é refletido nos valores obtidos pelo índice de justiça.

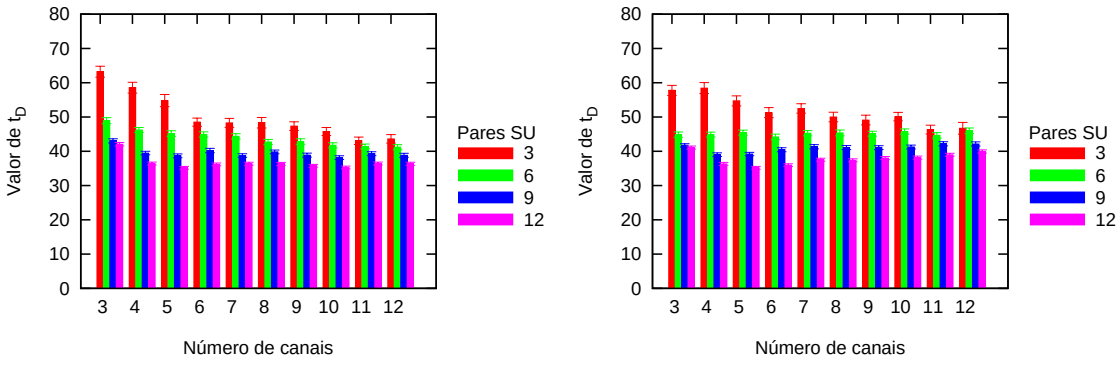
5.5 Consideração sobre parametrização do E^{2}S^3

A duração de uma época de decisão, t_D, influencia a acurácia das medições do canal e o tempo de resposta de maneira divergente em todos os mecanismos avaliados. Portanto, é importante escolher um valor que não seja muito pequeno para evitar um acurária muito baixa e nem muito grande para evitar que o mecanismo demore muito a reagir a alterações nas condições do canal.

No ES^3, foi adotado o valor de t_D igual a 30 para manter condições semelhantes de comparação com o *Q-Learning* [63]. Em relação ao E^{2}S^3, a duração de uma época, t_D, entra como um parâmetro adaptativo que é incrementado ou decrementado de acordo com o desempenho da vazão na época anterior. Esse controle permite que o mecanismo possa se adaptar e reagir de modo mais adequado com a dinamicidade do ambiente.

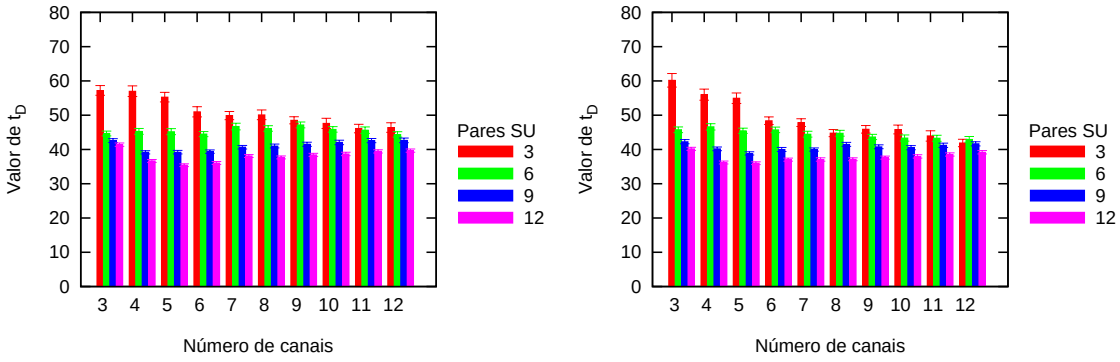
As Figuras 5.16(a), 5.16(b), 5.17(a) e 5.17(b) mostram a média dos valores máximos alcançados por t_D para 30 execuções. Em todas as figuras observa-se que o valor de t_D diminui quando a quantidade de pares aumenta para uma mesma quantidade de canais. Para a mesma quantidade de pares SU e com o aumento do número de canais o valor t_D não segue uma tendência pré-definida, variando o valor máximo alcançado. Isso acontece porque o acréscimo de canais implica em maior oferta de oportunidade que serão avaliadas e de acordo com a probabilidade de transmissão. Os SUs podem obter sucesso ou não, interferindo diretamente no valor alcançado por t_D, uma vez que a adaptação considera o desempenho obtido na época anterior.

A quantidade de épocas em uma geração, k, é um parâmetro que rege o processo evolutivo. Aplicado ao problema investigado nesse trabalho, esse parâmetro têm influência na classficação dos canais e, portanto, não pode ser muito alto sob pena de degradação no desempenho. Neste trabalho, foi adotado um valor k igual ao valor de uma época, t_d. Assim, k é igual 30 tentativas de transmissão no ES^3 e no *Q-Learning*. Esse valor se mostrou adequado a todos os cenários e aos mecanismos de seleção dinâmica cuja época de decisão é fixa. Isso porque os cenários dinâmicos possuem intervalos pré-definidos depois do qual o melhor canal é alterado. Por isso, o processo evolutivo deve acontecer



(a) *Cenário estático.* (b) *Cenário dinâmico A.*

Figura 5.16: *Média do valor máximo de t_D em cenários com menor dinamicidade.*



(a) *Cenário dinâmico B.* (b) *Cenário dinâmico C.*

Figura 5.17: *Média do valor máximo de t_D em cenários com maior dinamicidade.*

um número suficiente de gerações naquele intervalo para que o melhor canal possa ser encontrado.

A Tabela 5.8 mostra em relação a cada cenário a quantidade de gerações existentes de acordo com a parametrização do ES^3 e do *Q-Learning*. No dinâmico A, 666 gerações a cada intervalo de 20 mil tentativas de transmissão, no dinâmico B e C, 133 gerações a cada intervalo de 4 mil tentativas de transmissão. O valor adotado permitiu uma quantidade de gerações adequada para cenários dinâmicos.

Caso k adotasse valores maiores, a quantidade de gerações por intervalos poderia não ser suficiente para que o mecanismo encontrasse o melhor canal durante aquele intervalo. Por exemplo, no caso dos cenários dinâmicos B e C, cujo intervalo de troca do melhor canal é de 4 mil transmissões, para k igual a 5, haveria 150 tentativas de transmissões compondo uma geração, valor cinco vezes maior que a parametrização adotada. Existem, portanto, 26 gerações por intervalo. Esse valor representa quase 3% do total de tentativas de transmissão de todo o intervalo, enquanto que na parametrização

Tabela 5.8: *Parâmetro gerações em cada cenário.*

Cenários	Estático	Dinâmico A	Dinâmico B	Dinâmico C
Quantidade de intervalos	1	5	25	
Quantidade de transmissões por intervalo	100000	20000	4000	
k	1			
t_d	30			
Gerações por intervalo	3333	666	133	
Quantidade de gerações	3333			
Quantidade de transmissões	100000			

original esse valor é inferior a 1%.

Em relação ao E^2S^3, o parâmetro k também possui valor de uma época de decisão, t_D. Porém, como t_d é um parâmetro adaptativo, a quantidade de gerações varia a cada execução.

5.6 Conclusão

Neste capítulo, foram apresentados os resultados com relação ao número de sucessos e canais visitados para os mecanismos de seleção de espectro com o objetivo de mostrar um novo algoritmo que otimizasse o uso do espectro. Todos os experimentos foram realizados em uma plataforma de simulação, cuja corretude foi avaliada em relação a um mecanismo de referência de escolha ótima. Os cenários elaborados serviram para avaliar os mecanismos em condições semelhantes.

Além da corretude, também foi apresentada uma análise que mostrou que o uso da estratégia evolutiva pode maximizar o desempenho do mecanismo de seleção de espectro em relação a uma seleção aleatória.

Todas as avaliações feitas neste capítulo mostram que o algoritmo baseado em estratégia evolutiva consegue promover melhor uso do espectro e as diferenças de desempenho obtidas são toleráveis.

A melhoria trazida para o segunda versão do algoritmo baseado em estratégia evolutiva proporcionou a obtenção de um mecanismo adaptado a ambientes dinâmicos e com maior quantidade de dispositivos concorrendo ao uso do espectro.

CAPÍTULO 6

Conclusão e perspectiva de trabalhos futuros

Este trabalho teve como tema a área de rádios cognitivos. Durante o levantamento bibliográfico, artigos relacionados ao assunto serviram de referência e investigação de problemas em aberto, mostrando várias oportunidades de pesquisa. Alguns problemas relacionados às funções dos rádios cognitivos foram elencados e entre eles, a seleção dinâmica de espectro foi escolhida.

Uma vez selecionado o problema, as atividades foram voltadas para a busca de soluções já consolidadas em pesquisas dentro da área. Dessas soluções, a que mais se destacou foi o *Q-Learning*, pelos resultados alcançados e ampla difusão.

A partir de então, esse algoritmo foi estudado, bem como a área a qual pertence, a de Aprendizado por Reforço. No decorrer dos estudos, outra área, a Computação Evolutiva surgiu com um conjunto de algoritmos que mostraram potencial para problemas de otimização. Desse conjunto, a Estratégia Evolutiva foi a técnica que mais reuniu características com potencial para resolver o problema em estudo deste trabalho.

O trabalho resultou no desenvolvimento das seguintes atividades:

- Estudo sobre técnicas de Aprendizado por Reforço com foco no *Q-Learning*, a fim de entender o comportamento e propor melhorias para o uso na seleção dinâmica de espectro.
- Modelagem do problema em termos estocásticos com a finalidade de aplicação e avaliação das técnicas selecionadas.
- Construção da plataforma de simulação que abrangesse o modelo do problema e pudesse aplicar as técnicas em condições semelhantes de avaliação.
- Estudo do comportamento do *Q-Learning* e seus parâmetros para identificar possíveis correlações entre eles diante de cenários pré-definidos.
- Estudos para aperfeiçoamento do cálculo do valor-Q do *Q-Learning* no sentido de proporcionar rápida convergência.
- Estudo sobre Estratégias Evolutivas, técnica que reunia características que a tornava candidata em potencial para aplicação no problema investigado.
- Elaboração de um algoritmo concorrente baseado em Estratégia Evolutiva, com menor combinação de configuração devido a diminuição da quantidade de parâ-

metros em relação ao *Q-Learning*. A segunda versão agregou a característica de auto-adaptação dos parâmetros para aumentar sua autonomia e adequação para ambientes de maior concorrência.

Durante a realização das atividades do trabalho, foram encontradas as seguintes dificuldades:

- Melhoria do *Q-Learning*. A ideia inicial era de promover melhoria no cálculo do valor-Q, partindo da média móvel empregada pelo *Q-Learning* para o cálculo baseado na Suavização Exponencial Dupla [1], alternativa que não foi bem sucedida e logo abandonada.
- Processo de concepção do novo algoritmo. Construção de um algoritmo concorrente requer profundo entendimento do problema e do algoritmo concorrente. Para guiar o projeto e a análise do novo algoritmo demandou tempo precioso que se tivesse sido mal conduzido poderia comprometer o desenvolvimento do trabalho.
- Habilidade com programação. Essa foi a principal dificuldade desde o início do trabalho e que impediu a implementação das propostas dentro do *Network Simulator* (ns-3) [2]. Por isso, foi construída uma plataforma de simulação própria, que utilizasse uma linguagem de programação com curva de aprendizado mais rápida que a linguagem utilizada pelo ns-3.

As perspectivas de trabalhos futuros são descritas a seguir:

- Implementação do E^2S^3 no ns-3. Esta é uma plataforma mais próxima do real comportamento das tecnologias de comunicação atualmente empregadas.
- Embora o E^2S^3 tenha se mostrado bastante competitivo, ele ainda perde em alguns cenários e quando há bastante concorrência. Para suprir essa deficiência, seria necessário buscar melhorias no projeto do algoritmo poderiam proporcionar maior ganho.
- Implementação e avaliação do E^2S^3 em dispositivos reais. O mecanismo pode melhorar o desempenho dos dispositivos cognitivos tradicionais construídos sobre rádios definidos por software e também pode ser empregado em equipamentos baseados 802.11af [13] ou até mesmo em 802.11 convencionais.
- Uso de técnicas de previsão de séries temporais para auxiliar na seleção de oportunidades de transmissão e de outras distribuições para mutação.

Referências Bibliográficas

[1] **Engineering statistics handbook**. http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc433.htm.

[2] **Network simulator**. http://www.nsnam.org/.

[3] AKBAR, I. A.; TRANTER, W. H. **Dynamic spectrum allocation in cognitive radio using hidden Markov models: Poisson distributed case**. In: *IEEE SoutheastCon (SECON)*, p. 196–201, 2007.

[4] AKYILDIZ, I. F.; LEE, W.-Y.; VURAN, M. C.; MOHANTY, S. **NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey**. *Computer Networks*, 50:2127–2159, 2006.

[5] BACCHUS, R.; FERTNER, A.; HOOD, C.; ROBERSON, D. **Long-term, wide-band spectral monitoring in support of dynamic spectrum access networks at the iit spectrum observatory**. In: *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, p. 1–10, 2008.

[6] Back, T.; Fogel, D. B.; Michalewicz, Z., editors. **Handbook of Evolutionary Computation**. IOP Publishing Ltd., Bristol, UK, UK, 1st edition, 1997.

[7] BÄCK, T.; HOFFMEISTER, F.; SCHWEFEL, H.-P. **A survey of evolution strategies**. In: *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, p. 2–9. Morgan Kaufmann, 1991.

[8] BARBOSA, C. S.; BORGES, V. C. M.; CORREA, S.; CARDOSO, K. V. **An evolution-inspired algorithm for efficient dynamic spectrum selection**. In: *ICOIN*, p. 175–180, 2013.

[9] BARBOSA, C. S.; CORREA, S.; SALVINI, R.; CARDOSO, K. V. **Uma Estratégia Evolutiva para a Seleção Dinâmica de Espectro**. *II Workshop de Redes de Acesso em Banda Larga*, p. 59–72, Maio 2012.

[10] BLASCHKE, V.; JAEKEL, H.; RENK, T.; KLOECK, C.; JONDRAL, F. K. **Occupation measurements supporting dynamic spectrum allocation for cognitive radio design**. In: *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, p. 50–57, 2007.

[11] CHEN, X.; HUANG, J.; LI, H. **Adaptive channel recommendation for dynamic spectrum access**. In: *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symp. on*, p. 116–124, 2011.

[12] CHENG, X.; JIANG, M. **Cognitive Radio Spectrum Assignment Based on Artificial Bee Colony Algorithm**. In: *IEEE International Conference on Communication Technology (ICCT)*, 2011.

[13] DHOPE, T. S.; SIMUNIC, D. **Analyzing the Performance of Spectrum Sensing Algorithms for IEEE 802.11 af Standard in Cognitive Radio Network**. In: *Studies in Informatics and Control*, p. 93–100, 2012.

[14] DO, C.; TRAN, N.; HONG, C. S. **Throughput maximization for the secondary user over multi-channel cognitive radio networks**. In: *Information Networking (ICOIN), 2012 International Conference on*, p. 65–69, 2012.

[15] DOERR, C.; SICKER, D. C.; GRUNWALD, D. **Dynamic Control Channel Assignment in Cognitive Radio Networks using Swarm Intelligence**. In: *IEEE Global Telecommunications Conference (GLOBECOM)*, p. 1–6, 2008.

[16] DUAN, L.; GAO, L.; HUANG, J. **Contract-based cooperative spectrum sharing**. In: *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, p. 399–407, 2011.

[17] EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. SpringerVerlag, 2003.

[18] ELIAS, J.; MARTIGNON, F.; CAPONE, A.; ALTMAN, E. **Non-cooperative spectrum access in cognitive radio networks: A game theoretical model**. *Comput. Netw.*, 55(17):3832–3846, 2011.

[19] ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. Wiley Publishing, 2nd edition, 2007.

[20] FELICE, M. D.; CHOWDHURY, K. R.; WU, C.; BONONI, L.; MELEIS, W. **Learning-Based Spectrum Selection in Cognitive Radio Ad Hoc Networks**. In: *International Conference on Wired/Wireless Internet Communications (WWIC)*, 2010.

[21] FENG, X.; DAIMING, Q.; GUANGXI, Z.; YANCHUN, L. **Smart Channel Switching in Cognitive Radio Networks**. In: *Int. Congress on Image and Signal Processing (CISP)*, p. 1–5, 2009.

[22] FOGEL, D. B. **Evolutionary computation - toward a new philosophy of machine intelligence.** IEEE, 1995.

[23] GHASEMI, A.; SOUSA, E. S. **Collaborative Spectrum Sensing for Opportunistic Access in Fading Environments**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 131–136, 2005.

[24] GURNEY, D.; BUCHWALD, G.; ECKLUND, L.; KUFFNER, S. L.; GROSSPIETSCH, J. **Geo-Location Database Techniques for Incumbent Protection in the TV White Space**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 1–9, 2008.

[25] HARROLD, T.; CEPEDA, R.; BEACH, M. **Long-term measurements of spectrum occupancy characteristics**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 83–89, 2011.

[26] HOLLAND, O.; CORDIER, P.; MUCK, M.; MAZET, L.; KLOCK, C.; RENK, T. **Spectrum power measurements in 2g and 3g cellular phone bands during the 2006 football world cup in germany**. In: *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, p. 575–578, 2007.

[27] HÖYHTYÄ, M.; POLLIN, S.; MÄMMELÄ, A. **Performance Improvement with Predictive Channel Selection for Cognitive Radios**. In: *First International Workshop on Cognitive Radio and Advanced Spectrum Management (CogART)*, p. 1–5, 2008.

[28] III, J. M. **Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio**. PhD thesis, Royal Institute of Technology (KTH), 2000.

[29] III, J. M. **Software Radio: Wireless Architecture for the 21st Century**. Wiley-Blackwell, 2000. ISBN 978-0471384922.

[30] ISLAM, M.; KOH, C.; OH, S.; QING, X.; LAI, Y.; WANG, C.; LIANG, Y.-C.; TOH, B.; CHIN, F.; TAN, G.; TOH, W. **Spectrum survey in singapore: Occupancy measurements and analyses**. In: *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, p. 1–7, 2008.

[31] JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling**. Wiley- Interscience, New York, NY, 1991.

[32] JIA, J.; ZHANG, Q.; ZHANG, Q.; LIU, M. **Revenue generation for truthful spectrum auction in dynamic spectrum access**. In: *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, p. 3–12. ACM, 2009.

[33] JONG, K. A. D. **Evolutionary computation - a unified approach.** MIT Press, 2006.

[34] KIM, H.; SHIN, K. G. **Fast Discovery of Spectrum Opportunities in Cognitive Radio Networks**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 1–12, 2008.

[35] KIM, H.; SHIN, K. G. **In-band spectrum sensing in cognitive radio networks: energy detection or feature detection?** In: *ACM international conference on Mobile computing and networking (MobiCom)*, 2008.

[36] KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. MIT Press, Cambridge, MA, USA, 1992.

[37] KRAUSE, J.; CORDEIRO, J. A.; LOPES, H. S. **Comparação de métodos de computação evolucionária para o problema da mochila multidimensional**. In: Lopes, H. S.; de Abreu Rodrigues, L. C.; Steiner, M. T. A., editors, *Meta-Heurísticas em Pesquisa Operacional*, chapter 6, p. 87–98. Omnipax, Curitiba, PR, 1 edition, 2013.

[38] LAMARCK, J.-B. **Philosophie zoologique**. Paris :Duminil-Lesueur,. http://www.biodiversitylibrary.org/bibliography/50194.

[39] LI, H. **Multi-agent Q-learning of channel selection in multi-user cognitive radio systems: a two by two case**. In: *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, p. 1893–1898, 2009.

[40] LI, H. **Customer reviews in spectrum: Recommendation system in cognitive radio networks**. In: *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, p. 1–9, 2010.

[41] LINDEN, R. **Algoritmos Genéticos**. Ciência Moderna, 3rd edition, 2012.

[42] MALANCHINI, I.; CESANA, M.; GATTI, N. **On Spectrum Selection Games in Cognitive Radio Networks**. In: *IEEE Global Telec. Conference (GLOBECOM)*, p. 1–7, 2009.

[43] MARTIAN, A.; MARCU, I.; MARGHESCU, I. **Spectrum Occupancy in an Urban Environment: A Cognitive Radio Approach**. In: *Advanced International Conference on Telecommunications (AICT)*, p. 25–29, 2010.

[44] MITCHELL, T. M. **Michine Learning**. McGraw-Hill Science/Engineering/Math, 1997.

[45] MITOLA, J.; MAGUIRE, G. Q. **Cognitive Radio: Making Software Radios More Personal**. *IEEE Personal Comm.*, 6(4):13–18, 1999.

[46] MURTY, R.; CHANDRA, R.; MOSCIBRODA, T.; BAHL, P. V. **SenseLess: A Database-Driven White Spaces Network**. *IEEE Transactions on Mobile Computing*, 2012.

[47] PUTERMAN, M. L. **Markov Decision Processes: Discrete Stochastic Dynamic Programming**. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[48] REN, Y.; DMOCHOWSKI, P.; KOMISARCZUK, P. **Analysis and Implementation of Reinforcement Learning on a GNU Radio Cognitive Radio Platform**. In: *Int. Conf. on Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM)*, p. 1–6, 2010.

[49] SCHWEFEL, H. P. **Evolutionsstrategie und numerische optimierung**. PhD thesis, Technical University of Berlin, 1975.

[50] SCHWEFEL, H. P. **Numerical Optimization of Computer Models**. John Wiley & Sons, Inc., New York, NY, USA, 1981.

[51] SHU, T.; KRUNZ, M. **Coordinated Channel Access in Cognitive Radio Networks: A Multi-Level Spectrum Opportunity Perspective**. In: *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, p. 2976–2980, 2009.

[52] SUTTON, R. S.; BARKO, A. G. **Reinforcement Learning**. Cambridge: MIT Press, 1998.

[53] TAHER, T. M.; BACCHUS, R. B.; ZDUNEK, K. J.; DENNIS A, R. **Long-term spectral occupancy findings in Chicago**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 100–107, 2011.

[54] TANDRA, R.; SAHAI, A.; MISHRA, S. **What is a spectrum hole and what does it take to recognize one?** *Proceedings of the IEEE*, 97(5):824–848, may 2009.

[55] TANDRA, R.; SAHAI, A.; VEERAVALLI, V. V. **Space-time metrics for spectrum sensing**. In: *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 1–12, 2010.

[56] VIDAL, J. R.; PLA, V.; GUIJARRO, L.; MARTINEZ-BAUSET, J. **Flexible dynamic spectrum allocation in cognitive radio networks based on game-theoretical mechanism design**. In: *International IFIP TC 6 conference on Networking (NETWORKING)*, p. 164–177, 2011.

[57] VUCEVIC, N.; AKYILDIZ, I. F.; PEREZ-ROMERO, J. **Dynamic cooperator selection in cognitive radio networks**. *Ad Hoc Networks*, 10(5):789–802, 2012.

[58] WATKINS, C. J. C. H. **Learning from Delayed Rewards**. PhD thesis, Cambridge University, 1989.

[59] WELLENS, M.; MÄHÖNEN, P. **Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model**. In: *International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops (TridentCom)*, p. 1–9, 2009.

[60] WU, Z.; CHENG, P.; WANG, X.; GAN, X.; YU, H.; WANG, H. **Cooperative Spectrum Allocation for Cognitive Radio Network: An Evolutionary Approach**. In: *Communications (ICC), 2011 IEEE Int. Conf.*, p. 1–5, 2011.

[61] XU, D.; JUNG, E.; LIU, X. **Optimal bandwidth selection in multi-channel cognitive radio networks: How much is too much?** In: *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, p. 1–11, 2008.

[62] YAU, K.-L. A.; KOMISARCZUK, P.; TEAL, P. D. **A Context-aware and Intelligent Dynamic Channel Selection Scheme for Cognitive Radio Networks**. In: *Int. Conference on Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM)*, p. 1–6, 2009.

[63] YAU, K.-L. A.; KOMISARCZUK, P.; TEAL, P. D. **Enhancing network performance in Distributed Cognitive Radio Networks using single-agent and multi-agent Reinforcement Learning**. In: *Conference Local Computer Networks (LCN)*, p. 152–159, 2010.

[64] YAU, K.-L. A.; TEAL, P. D.; KOMISARCZUK, P. **Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues**. *Journal of Network and Computer Applications*, 35(1):253–267, 2012.

[65] ZHAO, J.; ZHENG, H.; YANG, G.-H. **Distributed coordination in dynamic spectrum allocation networks**. In: *New Frontiers in Dynamic Spectrum Access Networks. IEEE Int. Symp. on*, p. 259–268, 2005.

[66] ZHAO, Q.; TONG, L.; SWAMI, A.; CHEN, Y. **Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A pomdp framework**. *Selected Areas in Communications, IEEE Journal on*, 25(3):589–600, 2007.

APÊNDICE A

Algoritmos Genéticos, Programação Genética e Programação Evolutiva

A.0.1 Algoritmos genéticos

Algoritmos Genéticos é uma das principais técnicas dentro da Computação Evolutiva. Seu mecanismo de atuação baseia-se nos processos de evolução genética observados nos organismos biológicos, cuja fundamentação tem base na seleção natural das espécies, proposta por Charles Darwin. Seus fundamentos teóricos foram desenolvidos e propostos por John Holland e popularizou-se nos anos 80 [17].

A referência para atuação desses algoritmos é produzir novos cromossomos com propriedades genéticas superiores às de seus antecedentes. Essa ideia dentro do contexto biológico pode ser traduzida, de forma genérica, em gerar soluções eficientes para um problema. Assim, a partir de um conjunto de soluções atuais, são geradas novas soluções superiores às antecedentes, sob algum critério previamente estabelecido.

O principal operador genético é a recombinação e a mutação geralmente possui baixa probabilidade. De forma sucinta, esses operadores sobre indivíduos de representação binária de comprimento fixo funcionam do seguinte modo [41]:

- Mutação - modificação do símbolo presente em uma posição do cromossomo. A probabilidade de ocorrência de mutação em geral é da ordem de $1/l$, onde l é número de bits do cromossomo.
- Recombinação - ocorre em um ponto do cromossomo. Através de um esquema de seleção, dois indivíduos são escolhidos e de acordo com uma probabilidade, são submetidos à operação de recombinação. Uma posição do cromossomo é sorteada e o material genético dos pais é recombinado gerando dois descendentes. Outras variações deste operador também podem ser utilizadas, tais como, recombinação de dois pontos e recombinação uniforme.
- Seleção - a probabilidade de seleção de um indivíduo da população vir a ser selecionado é proporcional à sua aptidão relativa.

A.0.2 Programação genética

A Programação Genética foi desenvolvida por John Koza e consiste em realizar uma busca no espaço de possíveis programas computacionais com a finalidade de escolher o melhor deles para atingir um determinado objetivo. Portanto, essa é uma técnica de geração automática de programas, em que o programa induzido deva ser capaz de satisfazer as especificações de comportamento previamente definidas [36].

O fundamento dessa técnica é baseado na combinação dos conceitos de seleção natural, de operadores de reprodução, de busca heurística e da teoria de compiladores. Dessa forma, os programas são representados como árvores sintáticas e são formados a partir do processo evolutivo e da aplicação dos operadores genéticos à população. O operador de reprodução apenas seleciona um programa e o copia para a próxima geração e a taxa de mutação utilizada é baixa.

A etapa de seleção é realizada através da avaliação da aptidão dos programas [17]. Para isso, é fornecido um conjunto de casos para treinamento, com valores de entrada e saída a serem aprendidos. Os resultados obtidos são confrontados com valores esperados e, dessa forma, a avaliação considera a proximidade da resposta do programa e o valor de saída esperado.

A.0.3 Programação evolutiva

A Programação Evolutiva tem como objetivo desenvolver máquinas de estado finitos com a finalidade de prever eventos baseado nas observações anteriores. Essa é uma máquina abstrata que transforma uma sequência de dados de entrada em uma sequência de dados de saída cuja transformação depende de regras de transição pré-determinadas [19].

Atualmente, para representação dos indivíduos adota-se vetores de números reais e o processo de reprodução utiliza apenas a mutação. Cada genitor gera apenas um filho através de mutação que tipicamente aplica probabilidade uniforme. Quando o indivíduo é uma máquina de estado, a mutação implica em uma mudança aleatória em cinco possíveis itens: mudança de um dado de saída, de uma regra de transição, ou de uma regra de transição inicial, inclusão ou exclusão de uma regra de transição.

O processo de seleção ocorre como uma série de torneios entre subgrupos de indivíduos dentro da população. Cada indivíduo da população é avaliado em relação a outros indivíduos escolhidos aleatoriamente. Para cada comparação é marcado um vencedor, sendo esses que comporão a próxima geração.