



Eduardo Dessupoio Moreira Dias

**Modelo de Neuro Co-Evolução com Inspiração Quântica
Aplicado a Problemas de Coordenação**

Tese de Doutorado

Tese apresentada ao programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio como parte dos requisitos parciais para obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Profa. Marley Maria Bernardes Rebuszi Vellasco

Co-orientador: Prof. André Vargas Abs da Cruz

Rio de Janeiro

Abril de 2016



Eduardo Dessupoio Moreira Dias

**Modelo de Neuro Co-Evolução com Inspiração Quântica
Aplicado a Problemas de Coordenação**

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

Profa. Marley Maria Bernardes Rebuszi Vellasco
Orientadora
PUC-RIO

Prof. André Vargas Abs da Cruz
Co-Orientador
UEZO

Prof. Leonardo Alfredo Forero Mendoza
UERJ

Dr. Marco Aurélio Botelho da Silva
Departamento de Engenharia Elétrica - PUC-Rio

Profa. Karla Tereza Figueiredo Leite
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Fabiano Saldanha Gomes de Oliveira
UERJ

Prof. Paulo Fernando Ferreira Rosa
IME

Prof. Márcio da Silveira Carvalho
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 28 de abril de 2016

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

Eduardo Dessupoio Moreira Dias

Graduado em Licenciatura em Matemática pela UFJF, 2004. Mestre em Economia Aplicada pela Universidade Federal de Juiz de Fora (UFJF) em 2008. Doutorando em Engenharia Elétrica na Pontifícia Universidade Católica do Rio de Janeiro. Lecionou de 2006-2008 no Departamento de Matemática da UFJF e desde 2008 é professor efetivo do Instituto Federal do Rio de Janeiro (IFRJ), lecionando conteúdos nas Áreas de Matemática e Estatística. Atualmente, desenvolve pesquisas na Área de inteligência artificial.

Ficha Catalográfica

Dias, Eduardo Dessupoio Moreira

Modelo de neuro co-evolução com inspiração quântica aplicado a problemas de coordenação / Eduardo Dessupoio Moreira Dias; orientadora: Marley Maria Bernardes Rebuszi Vellasco; co-orientador: André Vargas Abs da Cruz. – 2016.

118 f.; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2016.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Co-evolução. 3. Neuro-evolução. 4. Inspiração quântica. 5. Coordenação. I. Vellasco, Marley Maria Bernardes Rebuszi. II. Cruz, André Vargas Abs da. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Ao meu filho Raul,
minha maior inspiração.

Agradecimentos

À PUC-Rio e seus funcionários, pela oportunidade e apoio concedidos.

Aos meus orientadores, professores Marley e André, pelo tempo que me foi dedicado, pelas cobranças feitas e pelos conselhos transmitidos, fundamentais para a conclusão e qualidade desta tese.

Aos demais professores com os quais tive a oportunidade de aprender durante esta trajetória.

Aos companheiros de PUC-Rio, destacando-se Marco e Harold.

Aos amigos Lianto, Marcos e Ronaldo pelas estadias concedidas.

Aos amigos Guilherme, Moraes, Pedro e Victor, que me acompanharam durante boa parte desta jornada e contribuíram diretamente para a realização deste trabalho.

À minha maravilhosa família, apoio emocional para todas as horas e que tanto amo. Em especial, às mulheres da minha vida: Naty e Sussu.

Resumo

Dias, Eduardo Dessupoio Moreira; Vellasco, Marley Maria Bernardes Rebuzzi (Orientadora); da Cruz, André Vargas Abs da (Co-orientador). **Modelo de Neuro Co-evolução com Inspiração Quântica Aplicado a Problemas de Coordenação**. Rio de Janeiro, 2016. 118p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Em diversos problemas encontrados na literatura, se faz necessária alguma coordenação entre os agentes para que a tarefa seja realizada de forma ótima. Entretanto, pode ser difícil a obtenção desta coordenação por conta da quantidade e características dos agentes, dinâmica do ambiente e/ou complexidade da tarefa. O objetivo principal deste estudo é propor um modelo que possa se adaptar a problemas heterogêneos de coordenação e de dimensões elevadas, com aprendizado autônomo e que tenha convergência satisfatória, o qual foi denominado *Modelo de Neuro Co-Evolução com Inspiração Quântica (NCoQ)*. O modelo se utiliza dos paradigmas da física quântica e da co-evolução biológica, evoluindo concomitantemente sub-populações de indivíduos quânticos para obter ganhos de convergência. A representação dos indivíduos por pulsos quânticos consegue reduzir o número de indivíduos em cada população, além de ser a mais recomendada para a utilização de neuro-evolução por conta da representação real. Ressalta-se também a capacidade do modelo em obter de forma autônoma a melhor configuração de arquitetura para as redes neurais de cada agente, não exigindo do programador a escolha deste parâmetro. Foram propostos novos operadores quânticos de *crossover* e mutação que foram comparados na otimização de funções de diversas dimensões. Para testar o desempenho do modelo, foram desenvolvidas, em linguagem MATLAB, simulações para o problema presa predador, para o *benchmark multi-rover* de exploração de ambientes e uma simulação para cobertura telefônica. Foram feitas comparações com outros modelos neuro-evolutivos encontrados na literatura, tendo o modelo NCoQ apresentado os melhores resultados.

Palavras-chave

Co-Evolução; Neuro-evolução; Inspiração Quântica; Coordenação.

Abstract

Dias, Eduardo Dessupoio Moreira; Vellasco, Marley Maria Bernardes Rebuzzi (Advisor); da Cruz, Andr[e Vargas Abs da (Co-advisor). **Quantum Inspired Neuro Co-Evolution Model Applied to Coordination Problems**. Rio de Janeiro, 2016. 118p. DSc. Thesis – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Many problems in the literature require some coordination among agents so a specific task can be executed more efficiently. However, this coordination can be difficult because of the quantity and characteristics of the agents, environment dynamics and/or task complexity. The main contribution of this Thesis is the proposal of a model, called *Quantum Inspired Neuro Co-Evolution* (NCoQ), that can adapt to heterogeneous multi-agent problems in high dimensions utilizing self-learning and that has satisfactory convergence. The model is inspired in quantum physics and biological co-evolution paradigms and evolves concomitantly subpopulations of quantum individuals to get convergence gains. The representation of individuals for quantum functions is able to reduce the numbers of individuals in each population and it is the most recommended for real neuro-evolution representation. It's also important to point out the model capacity in self-finding the best architecture of the neural networks agents, not requiring an *a priori* definition of this parameter. New crossover and mutation quantum operators were also proposed and compared in functions optimization of multiple dimensions. To test the model performance, three MATLAB simulations were developed: prey-predator task, multi-rover task and cell phone coverage area simulation. Comparisons were made against others neuro-evolution models found in literature and the NCoQ model attained the best results.

Keywords

Coevolution; Neuroevolution; Quantum Inspired Model; Coordination.

Sumário

1 Introdução	13
1.1. Motivação	13
1.2. Objetivos	16
1.3. Contribuições	16
1.4. Descrição do trabalho	17
1.5. Organização do Trabalho	19
2 Fundamentos	20
2.1. Coordenação	20
2.1.1. Teoria dos Agentes	21
2.1.2. Características dos Sistemas Multi-Agente	23
2.1.3. Coordenação de agentes	25
2.1.3.1. Aprendizado em equipe x Aprendizado simultâneo	27
2.1.3.2. Especialização em Problemas de Coordenação	28
2.1.3.3. Avaliação da Coordenação	30
2.2. Neuro-Evolução e Co-Evolução Cooperativa	31
2.2.1. Neuro-Evolução	32
2.2.1.1. <i>Enforced Subpopulations Method</i> (ESP)	33
2.2.1.2. Neuro Evolution of Augmenting Topologies (NEAT)	35
2.2.2. Co-evolução Cooperativa	37
2.3. Algoritmos Genéticos com Inspiração Quântica	38
2.3.1. AEIQ-R	40
2.3.2. Avanços na modelagem quântica	43
3 Modelo de Neuro Co-evolução com Inspiração Quântica (NCoQ)	46
3.1. Modelo NCoQ	47
3.1.1. População Quântica	48
3.1.2. Time quântico e a avaliação dos indivíduos	50
3.1.3. Neuro-evolução clássica e o operador <i>lesion</i> multi-agente	51
3.2. Evolução Quântica	54
3.2.1. Desenvolvimento e testes de operadores quânticos	56
3.3. Resultados – Otimização de funções e Comparação de Operadores Quânticos	59
3.3.1. Resultados para a função Ackley 2 (F1)	62
3.3.2. Resultados para a função Colville (F2)	63
3.3.3. Resultados para a função Dolan (F3)	64
3.3.4. Resultados para a função Alpine 2 (F4)	66
3.3.5. Comparação dos operadores	67
4 Estudos de Casos	69
4.1. Presa-Predador	69
4.1.1. Metodologia	71
4.1.1.1. Comportamento da presa	72
4.1.1.2. Comportamento dos predadores	72
4.1.1.3. Dinâmica de treinamento	73
4.1.2. Resultados dos treinamentos com o modelo NCoQ	76
4.1.3. Resultados dos testes com o modelo NCoQ	80
4.1.4. Análise do modelo para outras variações no ambiente	84
4.1.4.1. Testes para ambiente discreto 21x21	85
4.1.4.2. Testes para ambiente 9x9 contínuo	85
4.1.5. Análise do modelo para co-evolução competitiva	87

4.2. <i>Multi-Rover</i>	89
4.2.1. Ambiente	90
4.2.2. Sondas (<i>Rovers</i>)	91
4.2.2.1. Sensores para detecção de rubis	91
4.2.2.2. Sensores de identificação de sondas	92
4.2.3. Controle de ação das sondas	93
4.2.4. Dinâmica de treinamento e Neuro-Evolução	94
4.2.5. Resultados – <i>Multi-Rover</i>	96
4.3. Otimização de Redes de Telefonia	99
4.3.1. Modelos de propagação Okumura-Hata	100
4.3.2. Dinâmica de otimização	101
4.3.3. Ambiente de simulação – cobertura móvel	103
4.3.4. Resultados – Simulação de Cobertura Móvel	104
5 Considerações Finais e Trabalhos Futuros	108
5.1. Futebol de Robôs	109
5.2. Robôs Humanóides	110
Referências Bibliográficas	111

Lista de Figuras

Figura 2-1 Esquema Típico de um Agente (REIS, 2003)	21
Figura 2-2 Dinâmica do aprendizado por reforço	26
Figura 2-3 Busca através de Algoritmos Evolutivos	27
Figura 2-4 Especialização Comportamental x Especialização Morfológica	29
Figura 2-5 Tipos de especialização.....	30
Figura 2-6 Dinâmica da neuro-evolução	32
Figura 2-7 Método ESP	34
Figura 2-8 Mapeamento do genótipo para o fenótipo.....	36
Figura 2-9 Crossover no modelo NEAT	36
Figura 2-10 Co-evolução cooperativa	38
Figura 2-11 Representação gráfica das probabilidades de se observar os valores 0 e 1 para um q-bit qualquer.....	39
Figura 2-12 Representação do gene quântico	41
Figura 2-13 Atualização da População Quântica	42
Figura 2-14 Listagem completa do algoritmo evolutivo com inspiração quântica usando representação real	43
Figura 2-15 <i>Cross Over</i> Quântico - <i>One Point</i>	44
Figura 2-16 <i>Quantum Inversion Mutation Operator</i>	44
Figura 2-17 Mutação Quântica Swap.....	45
Figura 3-1 Algoritmo de Co-evolução com destaque para os demais blocos de instrução	48
Figura 3-2 População Quântica formada pelas subpopulações de especialistas	49
Figura 3-3 Dinâmica de avaliação dos indivíduos quânticos	50
Figura 3-4 Arquitetura da rede neural recorrente	52
Figura 3-5 Representação matricial do indivíduo utilizado	53
Figura 3-6 Evolução da população quântica	56
Figura 3-7 Visualização do operador de <i>crossover</i> aritmético quântico proposto.....	58
Figura 3-8 Visualização do operador de mutação aritmética quântica proposto.....	59
Figura 3-9 Função Ackley 2	60
Figura 3-10 Erros durante a otimização da função F1.....	62

Figura 3-11 Erros durante a otimização da função F2.....	64
Figura 3-12 Erros durante a otimização da função F3.....	65
Figura 3-13 Erros durante a otimização da função F4.....	66
Figura 4-1 <i>Grid</i> ortogonal mostrando o esquema do Jogo da Presa/Predador...	70
Figura 4-2 Simulação sem coordenação dos predadores	70
Figura 4-3 Entrada e saída da rede neural utilizada.....	73
Figura 4-4 Dinâmica sequencial de uma caçada.....	74
Figura 4-5 Comparação entre cenários.....	75
Figura 4-6 Avaliação Média dos Times clássicos.....	77
Figura 4-7 Número de passos necessários para a captura durante a evolução .	78
Figura 4-8 Gráfico da evolução do fitness da melhor avaliação	79
Figura 4-9 Número de neurônios na camada recorrente durante o treinamento	79
Figura 4-10 Posições iniciais da presa na fase de testes	81
Figura 4-11 Dinâmica de avaliação para co-evolução competitiva	88
Figura 4-12 Resultado da co-evolução competitiva.....	88
Figura 4-13 Ambiente <i>Multi-Rover</i>	89
Figura 4-14 Disposição dos sensores nas sondas	91
Figura 4-15 Arquitetura da Rede Neural Recorrente para a aplicação <i>Multi-Rover</i>	94
Figura 4-16 Avaliação Média dos Indivíduos Clássicos (<i>Multi-Rover</i>)	97
Figura 4-17 Gráfico da evolução do <i>fitness</i> da melhor avaliação (<i>Multi-Rover</i>) ..	97
Figura 4-18 Comparação entre metodologias (<i>Multi-Rover</i>).....	98
Figura 4-19 Entradas e Arquitetura utilizada para as saídas da Potência Piloto de cada torre	102
Figura 4-20 Mapa do ambiente simulado	104
Figura 4-21 - Evolução da avaliação média das equipes	105
Figura 4-22 Evolução da melhor avaliação	106
Figura 4-23 Percentual de potência disponível	106
Figura 4-24 Comparação entre o modelo NCoQ e um modelo aleatório	107
Figura 5-1 Modelo de mão artificial (Ungureanu, 2008)	110

Lista de Tabelas

Tabela 3-1 Funções <i>benchmark</i> utilizadas	60
Tabela 3-2 Legendas dos operadores quânticos	61
Tabela 3-3 Parâmetros utilizados nos testes de otimização.....	61
Tabela 3-4 Resultados da otimização para a função F1	63
Tabela 3-5 Resultados da otimização para a função F2	64
Tabela 3-6 Resultados da otimização para a função F3	65
Tabela 3-7 Resultados da otimização para a função F4	66
Tabela 3-8 Desempenho médio das configurações testadas	67
Tabela 4-1 Parâmetros utilizados no treinamento	76
Tabela 4-2 Configuração dos parâmetros utilizados no modelo de Neuro- Evolução Clássica.....	81
Tabela 4-3 Configuração dos parâmetros utilizados no modelo AEIQ-R.....	82
Tabela 4-4 Resultados dos testes.....	82
Tabela 4-5 Desempenho de modelos RL em relação ao modelo NCoQ	84
Tabela 4-6 Resultado dos testes para <i>grid</i> 21x21	85
Tabela 4-7 Resultados dos testes para <i>grid</i> 9x9 contínuo.....	86
Tabela 4-8 - Tipos de rubis e os sensores requeridos para detecção	90
Tabela 4-9 Características dos sensores de rubi	92
Tabela 4-10 Parâmetros da simulação <i>Multi-Rover</i> (ambiente).....	95
Tabela 4-11 Parâmetros da simulação <i>Multi-Rover</i> (NCoQ)	96
Tabela 4-12 Comparação entre os modelos no experimento <i>Multi-Rover</i>	99
Tabela 4-13 Percentual da população coberta pela tecnologia 4G no Brasil Fonte: (ANATEL, 2015).....	100
Tabela 4-14 Parâmetros da simulação do modelo NCoQ para cobertura móvel	103
Tabela 4-15 Parâmetros do ambiente da simulação de cobertura móvel	104

1

Introdução

1.1

Motivação

Em diversas situações do cotidiano, se faz necessário algum tipo de coordenação. Entende-se por coordenação o ato de trabalhar em conjunto de forma harmoniosa, no sentido de atingir um acordo ou objetivo comum. Como exemplos, pode-se citar: o voo no formato em “V” de um bando de aves que diminui a resistência do ar e minimiza o desgaste do grupo; a construção de um grande edifício, possível apenas com a atuação coordenada de homens e máquinas; uma boa equipe de futebol, que depende, além dos talentos individuais, do entrosamento entre os jogadores.

Porém, em muitas aplicações reais como nas indústrias, na computação e na robótica, um grande número de agentes deve executar, no menor tempo possível, uma quantidade diversificada de tarefas, tendo a possibilidade de mudança de atuação de acordo com um imprevisto ocorrido ou uma mudança de objetivos. Tal complexidade limita a ação do projetista, sendo a análise das interações entre os atuadores, simuladas ou reais, um caminho elementar a ser tomado visando-se a encontrar uma solução satisfatória.

Dá-se o nome de Sistemas Multi-Agente (SMA) à área que estuda a interação entre os agentes em um ambiente, podendo ser encontrados na literatura estudos nas mais diversas aplicações como diagnósticos (Davidson *et al.*, 2006), gerenciamento de energias renováveis (Abdessalem & Karaa, 2014), monitoramento de sistemas (Mcarthur, Strachan & Jahn, 2004), restauração de sistemas elétricos (Nagata & Sasaki, 2002), automação de sub-estações (Buse *et al.*, 2003), controle de veículos aéreos (Salichon & Tumer, 2012), controle de tráfego (Purohit & Mantri, 2013), leilão de energia (Correa, Vellasco & Figueiredo, 2013) e futebol de robôs (Mendoza, Vellasco & Figueiredo, 2014).

A coordenação é uma característica fundamental de um sistema de múltiplos agentes que executam alguma atividade em um ambiente compartilhado (Goodwine & Antsaklis, 2010). Ela está intimamente relacionada ao compartilhamento de conhecimento entre os agentes e o seu principal objetivo é coordenar as ações de cada indivíduo para atingir o objetivo final do

sistema (Berndt & Herzog, 2011). Podem ocorrer diversos problemas em um SMA sem mecanismo de coordenação, tais como conflitos no acesso aos recursos disponíveis ou uso indevido, redundância na realização das tarefas e aumento do tempo de espera quando a atividade de um agente depende da realização de atividades por outros agentes (Lau & Reis, 2007). Assim, um bom sistema de coordenação deve evitar ou minimizar a ocorrência desses problemas, otimizando os recursos e o tempo para alcançar os objetivos (Sellner *et al.*, 2006).

Uma característica observável em diversos sistemas adaptativos é a necessidade de especialização dos agentes para que o comportamento ótimo coletivo seja obtido de maneira eficiente. Em sistemas ecológicos complexos, agentes especializados evoluíram ao longo do tempo com objetivo de diversificar a comunidade, a fim de se adaptarem melhor ao ambiente (Seligmann, 1999). Essas especializações ocorridas ao longo da evolução podem acontecer através de uma simples mudança de comportamento de um grupo específico (Bonabeau, Theraulaz & Deneubourg, 1996) ou, de uma forma mais complexa, através da alteração morfológica de alguns indivíduos da comunidade (Wensellers, Ratnieks & Billen, 2003). Por exemplo, as abelhas adaptam dinamicamente o seu comportamento recolhendo pólen, néctar ou água de acordo com a demanda da colônia (Calderone & Page Jr, 1988); já em certas colônias de cupins, pode-se observar castas morfológicamente especializadas (Noirot & Pasteels, 1987). A consequência destas especializações é uma divisão mais eficiente das tarefas entre os indivíduos, gerando benefícios para todo o grupo.

Em muitos casos, pelas complexidades do problema e da dinâmica do ambiente, não é possível obter uma estratégia de coordenação analiticamente, fazendo-se necessária uma interação entre os agentes e o ambiente para que ocorra o aprendizado, sem a interferência do programador. Exemplos desses casos podem ser encontrados na literatura mais recente: (Zhang & Zhao, 2013) desenvolveram um modelo de aprendizado por reforço para problemas multi-agente que utilizam grafos para decompor a função de avaliação global; (Quiñonez, Tostado & Sánchez, 2013) evoluíram neuro-controladores com arquiteturas e estratégias pré-definidas para deslocamento coordenado de robôs virtuais; (Colby, Knudson & Tumer, 2014) criaram um algoritmo de co-evolução cooperativa para controle de voo em ambientes dinâmicos.

Entretanto, a utilização de aprendizagem por reforço em um ambiente multi-agente de dimensão elevada pode apresentar um crescimento exponencial

de memória em relação à quantidade de agentes envolvidos na aplicação, visto que as funções de avaliação devem representar uma combinação dos estados e ações de todos os agentes em conjunto, e não de apenas um agente (Corrêa, 2011). A neuro-evolução é uma alternativa para esta questão, devido à capacidade de generalização das redes neurais, não exigindo um mapeamento de todo ambiente. Em contrapartida, pode esbarrar na dificuldade de convergência do algoritmo, dado que o número de genes a serem otimizados aumenta conforme o número de agentes envolvidos e a complexidade de arquitetura da rede, além da necessidade da avaliação de todos os indivíduos durante cada geração da evolução.

Outra limitação dos algoritmos evolutivos tradicionais é que, normalmente, não mantêm conhecimento normativo sobre o espaço de busca, ou seja, a informação contida no código genético de um indivíduo é pontual e não é compartilhada com os demais agentes. No caso de SMA, isso se torna ainda mais grave, pois a avaliação de um indivíduo depende, não apenas de suas características, mas, principalmente, de como o indivíduo interage com os demais agentes.

Uma alternativa para minimizar o problema do número de avaliações das soluções dos algoritmos evolutivos e a pontualidade da informação nos genes é o uso dos algoritmos genéticos com inspiração quântica (Cruz & Vellasco, 2010), (Cruz, 2007), (Han & Kim, 2000), (Han & Kim, 2004). Estes algoritmos se baseiam no princípio da sobreposição existente na Mecânica Quântica. Nesta teoria, um sistema físico existe parcialmente em todos os estados teoricamente possíveis antes de ser medido, ou seja, o bit quântico pode assumir qualquer um dos dois estados, 0 ou 1, com determinada probabilidade. Isto permite que um único indivíduo quântico possa representar todos os possíveis estados, permitindo uma redução no tamanho da população e, conseqüentemente, diminuindo o esforço computacional.

Mesmo num algoritmo evolutivo tradicional se faz necessária a atenção para a representação genética dos indivíduos a ser utilizada, pois fatores como o número de genes e as características representadas por estes influenciam diretamente no desempenho do algoritmo. A natureza da representação, binária ou real, também deve ser levada em consideração.

Outro fator que altera o desempenho de um algoritmo evolutivo é a escolha dos operadores genéticos a serem utilizados durante a evolução. Ao longo dos anos, diversos operadores, tanto de *crossover* quanto de mutação, foram desenvolvidos e combinados visando a repassar, para as gerações posteriores,

características capazes de melhorar a avaliação dos indivíduos, sem perder a diversidade da população.

1.2

Objetivos

Em função do exposto na seção anterior, o principal objetivo desse trabalho é desenvolver um modelo com inspiração quântica a ser aplicado em problemas de coordenação de equipes e que possua as seguintes características:

- 1) Autonomia de aprendizado: a atuação do programador fica limitada em tarefas de coordenação de maior complexidade como, por exemplo, a coordenação de uma equipe em um ambiente desconhecido. Por conta disso, é interessante que o aprendizado seja obtido a partir da interação dos agentes com o ambiente;
- 2) Aprendizado compartilhado: problemas de coordenação exigem sintonia entre os agentes. Dessa forma, o conhecimento deve ser compartilhado aos demais indivíduos, visando ganhos coletivos de desempenho;
- 3) Rápida convergência: diante da alta dimensionalidade presente em diversos problemas multi-agente, é fundamental que o novo modelo seja capaz de evoluir as soluções com um número menor de chamadas à função de avaliação das equipes;
- 4) Possibilidade de mudança da arquitetura: permite uma melhor adaptação do modelo a problemas SMA heterogêneos, onde os agentes possuem diferentes mecanismos de aprendizado, além de otimizar a arquitetura da rede durante o processo evolutivo.

1.3

Contribuições

A principal contribuição deste trabalho é o desenvolvimento de um novo modelo denominado *Modelo de Neuro Co-Evolução com Inspiração Quântica* (NCoQ) e que possui como características:

- 1) Neuro-evolução: permite que os agentes, de maneira independente, extraiam conhecimento através da interação direta com o ambiente e os demais agentes;
- 2) Aprendizado compartilhado: a co-evolução, de maneira implícita, faz com que o conhecimento armazenado seja repassado aos demais agentes, sendo fundamental para a coordenação dos mesmos;
- 3) Inspiração quântica: propicia o conhecimento normativo e a diminuição da população, aumentando a capacidade de busca do algoritmo;
- 4) Redes neurais auto-ajustáveis: a possibilidade de alteração das arquiteturas das redes minimiza a atuação do programador na escolha deste parâmetro. Também permite uma melhor adaptação do modelo a problemas SMA heterogêneos, onde os agentes podem vir a possuir diferentes mecanismos de aprendizado.

Além desta contribuição principal, podem-se destacar outras contribuições secundárias como:

- Proposição e testes de novos operadores quânticos de *crossover* e mutação usados na atualização da população quântica para problemas com representação real;
- Aplicação do NCoQ em problemas *benchmark*, como o problema presa-predador e a aplicação *multi-rover*;
- Aplicação do NCoQ em um problema real, referente à otimização da cobertura de uma rede de telefonia, na qual o modelo NCoQ atua no controle do sinal piloto das torres de transmissão de dados;
- Comparação entre o NCoQ e outros modelos encontrados na literatura.

1.4

Descrição do trabalho

Em modelos co-evolutivos cooperativos, os indivíduos são avaliados num contexto de equipe, não sendo mais a aptidão de cada indivíduo um valor absoluto dado por uma função de avaliação, mas sim um valor relativo baseado em seu desempenho no contexto dos demais indivíduos no mesmo processo de otimização.

Com o objetivo de diminuir os problemas da dimensionalidade elevada dos SMA e acelerar a convergência, no modelo NCoQ subpopulações de indivíduos

quânticos são evoluídas concomitantemente, mapeando as regiões mais promissoras do espaço de busca e gerando indivíduos com maiores afinidades entre si. Para que isso ocorra, a aptidão de um indivíduo é calculada selecionando indivíduos das outras populações e agrupando-os em uma equipe completa.

Por se tratar de um problema de otimização numérica, a utilização de representação binária não é a mais recomendada. Por conta disso, na evolução quântica será utilizado o algoritmo AEIQ-R, proposto por (Cruz, 2007), que utiliza genes com codificação real. Ressalta-se que foram necessárias pequenas adaptações no modelo original para o contexto multi-agente.

Uma característica desejável de um sistema adaptativo é que o mesmo seja capaz de propiciar a especialização de alguns agentes, necessária para uma atuação ótima da equipe a ser coordenada. A própria co-evolução já seria capaz de extrair dinamicamente comportamentos distintos dos agentes durante o processo evolutivo. Mas, assim como na natureza, em muitos problemas é necessária uma evolução morfológica dos indivíduos. Neste sentido, é fundamental que o processo evolutivo permita que os agentes envolvidos possam evoluir para redes neurais com arquiteturas distintas.

Para suprir esta demanda, o modelo NCoQ faz uso do operador *lesion* proposto por (Cruz, 2007). Este operador é responsável por modificar a arquitetura da rede de forma automática, fazendo a retirada ou o acréscimo de neurônios conforme a necessidade da evolução. Em problemas SMA heterogêneos, onde os agentes possuem naturezas e comportamentos ótimos diversos, a possibilidade de evoluções distintas de arquiteturas entre os agentes é altamente desejável, visto que a arquitetura da rede é um dos parâmetros que afeta diretamente o comportamento dos agentes.

Podemos, de forma geral, considerar que, para a realização do presente trabalho, quatro macro etapas foram realizadas:

- 1) Levantamento bibliográfico sobre os pilares teóricos do modelo proposto;
- 2) Desenvolvimento de um novo modelo com inspiração quântica para coordenação e aplicável a SMA, capaz de obter sua própria estrutura e de aprender dinamicamente, viabilizando aplicações em ambientes com espaços de estados grandes e/ou contínuos;
- 3) Desenvolvimento de novos operadores quânticos para problemas com representação real que foram testados na otimização de funções;

- 4) Desenvolvimento de um ambiente computacional em linguagem MATLAB dos *benchmarks* utilizados na avaliação do modelo proposto, permitindo a realização de simulações para demonstrar a potencialidade e a aplicabilidade do mesmo.

1.5

Organização do Trabalho

Esta tese possui mais cinco capítulos, além deste introdutório. No capítulo 2 é apresentada a revisão bibliográfica e um resumo dos fundamentos teóricos necessários para a compreensão deste trabalho.

No terceiro capítulo, o modelo NCoQ é apresentado detalhadamente, destacando-se a dinâmica de atualização da população quântica e da avaliação das equipes, bem como o estudo a respeito dos novos operadores quânticos, apresentados e testados para otimizar quatro funções de dimensões distintas. Ao final do capítulo, uma comparação entre as combinações dos operadores é feita.

O capítulo seguinte descreve em detalhes as aplicações simuladas para testar o modelo NCoQ. Na primeira parte é apresentada a problemática presa-predador utilizada para a validação do modelo, bem como os resultados obtidos. Em seguida, são apresentados os detalhes da aplicação *multi-rover*, seus resultados e a comparação com outros modelos encontrados na literatura. Na última aplicação, o modelo NCoQ é utilizado para controle do sinal piloto de torres de transmissão, visando maximizar a cobertura de uma rede de telefonia simulada.

Por fim, no quinto capítulo, são feitas as considerações finais e as sugestões para trabalhos futuros.

2

Fundamentos

2.1

Coordenação

Segundo (Malone & Crowston, 1994), coordenação é o ato de gerenciar dependências entre atividades. Essas dependências tendem a aumentar quando as atividades acontecem em um mesmo ambiente, decorrentes das ações de múltiplos agentes.

Há várias áreas que têm interesse em estudos de coordenação, as quais se podem citar como exemplos (Ossowski & García-Serrano, 1995) *apud* (Frozza, 2004):

- Sociologia: onde os sociólogos querem explicar como um mecanismo de coordenação atua em uma sociedade de pessoas;
- Economia: onde os economistas estudam o processo de marketing como um mecanismo de coordenação;
- Teoria organizacional: onde os teóricos organizacionais tentam explicar o comportamento coordenado de uma organização e também prever comportamentos futuros.

No campo da inteligência artificial, a coordenação pode ser vista como o processo onde um agente raciocina sobre a ação a ser tomada, considerando as possíveis ações dos demais agentes do ambiente, com o objetivo de fazer com que a comunidade atue de maneira coerente.

Segundo (Jennings, 1996), há três principais motivos para que as ações de múltiplos agentes sejam coordenadas:

- 1) *Quando existe dependência entre as ações dos agentes*: a interdependência ocorre quando os objetivos dos indivíduos se relacionam, seja quando a decisão de um agente interfere na decisão dos demais agentes (na construção de uma casa, decisões sobre o tamanho de um cômodo influenciam na canalização e na fiação), seja pela possibilidade de intromissão danosa entre os agentes (quando dois robôs tentam sair por uma única porta ao mesmo tempo, resultando em colisão e danos, além de bloquearem a passagem);

- 2) *Quando há a necessidade de se adaptar a restrições globais:* isso ocorre quando a solução desenvolvida por um grupo de agentes deve satisfazer obrigatoriamente determinada condição. Por exemplo, um sistema de tráfego aéreo que possui largura de banda limitada. Dessa forma, apenas soluções que respeitam as restrições são aceitas.
- 3) *Quando nenhum indivíduo tiver competência, recursos ou informações suficientes para resolver o problema por completo:* muitos problemas não podem ser resolvidos por indivíduos trabalhando de forma isolada, como a suspensão de uma carga muito pesada ou uma orquestra que toca uma sinfonia.

Como se pode ver, há várias situações onde é necessária a coordenação do que chamaremos de agentes. No item a seguir, tais entidades serão detalhadas.

2.1.1

Teoria dos Agentes

Podemos encontrar na literatura diversas formas de definir e classificar agentes, pois as mesmas estão associadas às funcionalidades e características do agente em questão.

(Russell & Norvig, 1995) entendem que um agente pode ser visto como aquilo que percebe seu entorno através de sensores e que atua sobre ele mediante seus executores, conforme pode ser visto na figura 2-1.

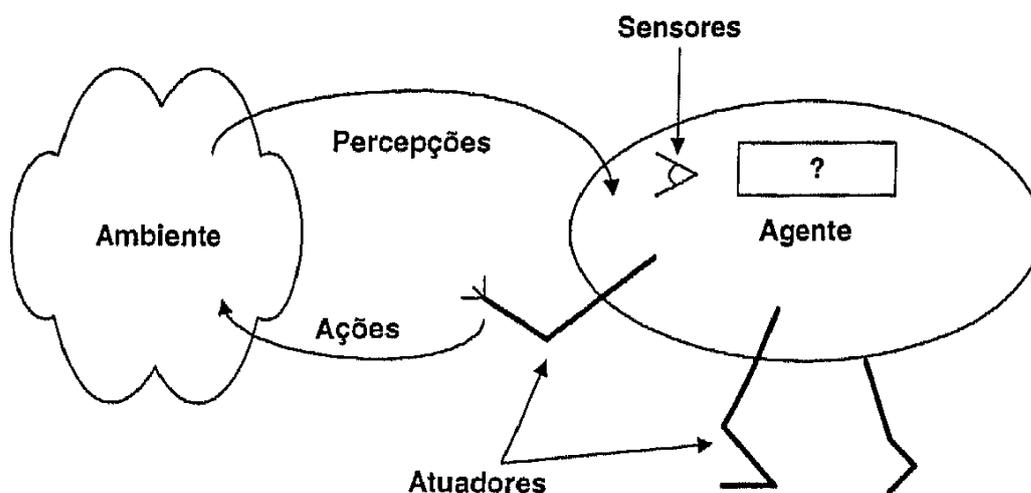


Figura 2-1: Esquema Típico de um Agente (REIS, 2003).

Segundo (Wooldridge, 1999), um agente inteligente é uma peça de hardware ou (mais usual) um sistema computacional baseado em software que goza das seguintes propriedades:

- Autonomia: os agentes operam sem a intervenção direta de humanos ou outros agentes e possuem algum tipo de controle sobre as suas ações e estado interno;
- Reatividade: os agentes têm a percepção do seu ambiente e respondem rapidamente às alterações que nele ocorrem;
- Pró-Atividade: os agentes não se limitam a agir em resposta ao seu ambiente. Eles são capazes de tomar a iniciativa e exibir comportamento direcionado para os objetivos;
- Habilidade Social: os agentes são capazes de interagir com outros agentes (e possivelmente com humanos) através de uma dada linguagem de comunicação de agentes.

Além dessas características, existem outras relacionadas especificamente ao comportamento humano, como o conhecimento, a crença, a intenção e a obrigação. Essas características são o que caracterizam a noção forte de agentes (Wooldridge & Jennings, 1995).

(Weiss, 1999) faz uma comparação entre objetos e agentes e conclui que, embora haja semelhanças, as diferenças são mais evidentes, principalmente, pelo fato dos agentes terem uma noção mais forte de autonomia em relação aos objetos e de apresentarem comportamentos mais adaptativos.

Nota-se que uma característica fundamental de um agente é a autonomia, entendendo-a como a capacidade de tomar ações de forma independente, sem a interferência do programador. Neste caso, o agente deve interagir com o ambiente e aprender a tomar as melhores decisões para atingir seus objetivos.

A metodologia seguida na construção de um agente determina a sua arquitetura (Russell & Norvig, 1995). Uma classificação geral para as diferentes arquiteturas dos agentes foi definida em (Wooldridge & Jennings, 1995):

- Arquiteturas Cognitivas: onde as decisões dos agentes são tomadas através de processos sofisticados de deliberação. O agente possui uma representação interna do mundo e um estado mental explícito que pode ser modificado por alguma forma de raciocínio simbólico;

- Arquiteturas Reativas: onde as decisões são tomadas com base em regras de reação ao estado atual do ambiente. A ação do agente ocorre em tempo real, em resposta a estímulos do ambiente captados por seus sensores ou a mensagens enviadas por outro agente;
- Arquiteturas Híbridas: que combinam as características das duas arquiteturas anteriormente citadas.

Neste estudo, as arquiteturas dos agentes terão características reativas por conta da utilização de redes neurais, já que estas receberão estímulos a partir da camada de entrada e, instantaneamente, produzirão a saída responsável pela ação do indivíduo. Tal arquitetura tem vantagens quando aplicada a cenários complexos e que envolvem muitos agentes, dada a sua simplicidade, o baixo custo computacional e a robustez contra falhas.

Porém, existem estudos onde há múltiplos agentes e assim, a ação tomada por cada um destes interferirá no ambiente e no retorno para todos os indivíduos do sistema. Logo, se faz necessária uma avaliação sobre esse comportamento coletivo. A área que estuda estes tipos de problemas é denominada Sistemas Multi-Agente (SMA).

2.1.2

Características dos Sistemas Multi-Agente

Os Sistemas Multi-Agente (SMA) constituem um campo relativamente novo. Embora o início da investigação nesta área tenha iniciado a partir dos anos 80, só em meados dos anos 90 ganhou notoriedade digna de destaque (Wooldridge, 2009). Ao longo dos últimos anos, a investigação no campo dos SMA tem sofrido um acentuado crescimento e, atualmente, está interessada, principalmente, no comportamento coletivo dos agentes, ou seja, como estes coordenam suas ações e compartilham seus conhecimentos, planos, objetivos e crenças com o objetivo de resolver problemas de maneira ótima.

SMA são sistemas compostos por múltiplos agentes que exibem comportamento autônomo, mas ao mesmo tempo interagem com os outros agentes presentes no sistema ou ambiente.

Os benefícios trazidos pela aplicação de SMA são diversos. Em primeiro lugar, através da computação paralela, vários agentes podem trabalhar em conjunto para melhor explorar a estrutura descentralizada de uma determinada tarefa e acelerar sua conclusão (Guestrin, Lagoudakis & Parr, 2002) (Kok *et al.*,

2005). Além disso, os SMA podem fornecer alto grau de escalabilidade, através da inclusão de novos agentes quando necessário, e ainda fazer com que agentes assumam as atividades de outros agentes em casos de falha (Busoni & Babuska, 2008).

Outras motivações para o uso de SMA são citadas por (Reis, 2003):

- Quando a dimensão do problema for demasiado elevada para poder ser resolvido por um único agente;
- Permitir a interconexão e interoperação de múltiplos sistemas legados, isto é, sistemas de gerações anteriores cuja manutenção do código já não é possível;
- Providenciar uma solução natural para problemas geograficamente e/ou funcionalmente distribuídos;
- Fornecer soluções para problemas em que os peritos, os conhecimentos ou as informações necessárias para a sua resolução, se encontram distribuídos;
- Permitir uma interface cooperativa homem-máquina mais natural em que ambos funcionam como agentes no sistema.

(Stone & Veloso, 2000) afirmam que os SMA potencialmente oferecem uma maior rentabilidade de recursos para problemas onde o conhecimento ou a atividade são distribuídos. Ainda há razões adicionais para a utilização de um SMA:

- O próprio domínio do problema, por exemplo, devido à distribuição espacial dos intervenientes;
- O paralelismo, atribuindo tarefas a diferentes agentes de forma que a execução seja mais rápida;
- A robustez, pois ao se utilizar diferentes agentes, elimina-se, desta forma, a existência de um ponto único de falha nos sistemas;
- A escalabilidade, permitindo o aumento dos agentes num determinado sistema aberto;
- A simplificação das tarefas individuais de programação, dividindo o problema global em vários sub-problemas;
- O estudo da inteligência individual e do comportamento social, pois os SMA permitem a inter-operacionalidade entre os agentes;

- A manutenção da privacidade da informação e conhecimentos individuais de cada agente.

As aplicações em SMA podem ser classificadas de diversas formas de acordo com o ambiente, homogeneidade de aprendizado, comunicação, tipo de tarefa, dentre outros. Alguns tipos de classificação podem ser vistos a seguir:

- Quanto ao ambiente: estático, se o ambiente não sofre alterações durante o processo, ou dinâmico, caso contrário;
- Homogeneidade de aprendizado: homogêneo, se todos os agentes aprendem da mesma forma, ou heterogêneo, se diferentes algoritmos de aprendizado são utilizados com o objetivo de obter comportamentos distintos dos agentes;
- Comunicação: se existe ou não alguma espécie de comunicação entre os agentes.
- Tipo da tarefa: totalmente competitivas, totalmente cooperativas ou mistas. No primeiro caso, os agentes competem entre si gerando, necessariamente, perdedores e vencedores. No segundo caso, os agentes buscam maximizar o retorno comum, sendo que os retornos são correlatos. No terceiro caso, não há qualquer restrição ao retorno dos agentes.

O foco do presente estudo está nos problemas que envolvem tarefas cooperativas. A seguir, serão apresentados estudos e técnicas que envolvem coordenação e que serviram de embasamento para o modelo proposto.

2.1.3

Coordenação de agentes

No contexto dos SMA, coordenação é o processo pelo qual um agente raciocina sobre suas ações locais e as ações dos outros para tentar garantir que a comunidade se comporte de forma coerente. Portanto, uma coordenação eficaz entre os agentes autônomos contribui para o aumento dos desempenhos dos mesmos e, conseqüentemente, para a qualidade das soluções encontradas.

Métodos estatísticos (Košinár & Kohut, 2010), raciocínio baseado em casos (Linghu & Chen, 2014), árvores de decisão (Hwang, Chen & Jiang, 2011), redes neurais (Shi & Shen, 2015), programação lógica indutiva (Komura, Mutoh

& Inuzuka, 2014) e inteligência de enxames (Júnior, 2008) são algumas das metodologias encontradas na literatura para a obtenção da coordenação de agentes. Porém, em especial, duas técnicas de aprendizado têm ocupado posição de destaque na literatura: aprendizado por reforço e algoritmos evolutivos. Tal relevância pode ser justificada por estas serem técnicas de aprendizado não supervisionado, ou seja, os agentes são capazes de aprender através da interação com o ambiente, não necessitando assim de um conhecimento prévio a respeito das soluções a serem encontradas.

No aprendizado por reforço, ou *Reinforcement Learning* (RL), o agente, através de um processo tentativa-e-erro, deve fazer o mapeamento estado-ação de modo a maximizar um sinal de recompensa. Em (Zhang & Zhao, 2013), (Hu, Gao & An, 2015) e (Vassiliades, Cleanthous & Christodoulou, 2011) podem ser encontrados maiores detalhes sobre o emprego desta metodologia em cenários multi-agente. A figura 2-2 ilustra o processo de aprendizado através desta metodologia.

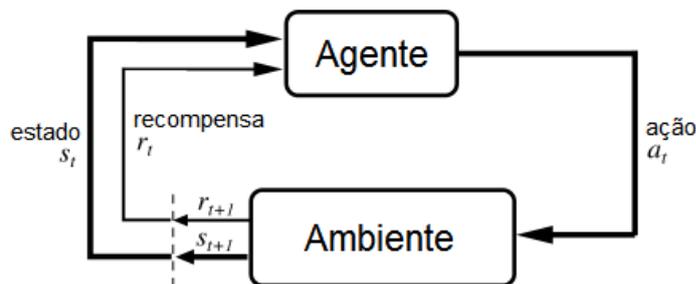


Figura 2-2: Dinâmica do aprendizado por reforço.

Entretanto, segundo (Bazzan, 2010), são três os problemas comumente encontrados por quem modela um problema de RL. O primeiro está ligado à atribuição temporal da recompensa (*temporal credit assignment*). A recompensa pode ser atribuída (ou percebida) com atraso, de modo que ela pode não ser necessariamente relacionada com a última ação realizada. O segundo problema é relacionado com o espaço de pares estado-ação. Os algoritmos de RL em geral exigem a visita frequente de cada um destes pares. Este problema é extremamente agravado em cenários multi-agente. O terceiro problema está ligado ao dilema exploração-aproveitamento (ou exploração-desfrute) que ocorre porque uma política é aprendida pelo agente ao mesmo tempo em que este explora seu ambiente.

Algoritmos evolutivos (Michalewicz, 1996), (Barros *et al.*, 2012) (Liu, Zhong & Jiao, 2010) utilizam-se de paradigmas da evolução de espécies para melhorar soluções iniciais a partir da interação dos agentes com o ambiente. Nesta metodologia, uma população de soluções iniciais é avaliada através de uma função objetivo que afere a adaptação das soluções ao ambiente. As melhores soluções são selecionadas para serem recombinadas e formarem a próxima geração de soluções. Tal ciclo é mantido durante certo número de gerações até que uma solução satisfatória seja encontrada. A figura 2-3 explica de forma resumida o ciclo de um algoritmo evolutivo.

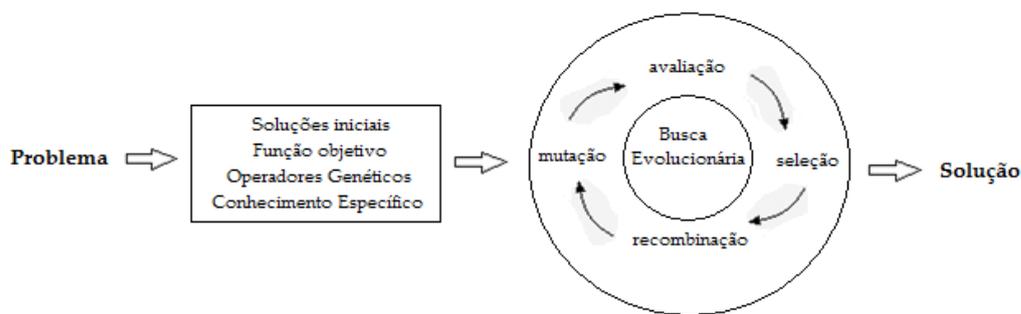


Figura 2-3: Busca através de Algoritmos Evolutivos.

2.1.3.1

Aprendizado em equipe x Aprendizado simultâneo

(Panait, 2006) defende que as abordagens cooperativas de aprendizagem se dividem em dois grupos distintos: aprendizado em equipe e aprendizado simultâneo.

No aprendizado em equipe, uma única entidade controla todos os atuadores, ou seja, um único agente controlador é responsável pelas ações da equipe. Exemplos desta técnica podem ser encontrados em (Su *et al.*, 2009) e (Wang & Liu, 2013).

O aprendizado em equipe se enquadra no aprendizado de máquina tradicional e pode apresentar problemas de escalabilidade quando o número de agentes do sistema aumenta. Por exemplo, se um agente A pode assumir 100 estados possíveis e um agente B pode assumir outros 100 estados, então a combinação de estados entre estes agentes é da ordem de 10000. Este aumento exponencial do número de estados em função do número de agentes pode tornar esta técnica proibitiva em alguns casos. Outra desvantagem

importante é a centralização de informação em um único agente, pois os recursos precisam estar disponíveis para este, o que pode ser oneroso em domínios onde os dados estão distribuídos.

Uma alternativa ao aprendizado em equipe é o aprendizado simultâneo. Nesta abordagem, múltiplos processos de aprendizagem acontecem concomitantemente, evoluindo de maneira independente e coordenada partes distintas da equipe. Além de não padecer dos problemas apontados no parágrafo anterior, o desenvolvimento independente dos agentes permite uma maior modularização das soluções e, conseqüentemente, numa maior possibilidade do uso de recursos computacionais e de minimização de falhas. Além disso, permite uma possibilidade maior da especialização de alguns agentes, característica desejável em diversas situações, conforme pode ser visto no próximo item.

2.1.3.2

Especialização em Problemas de Coordenação

A especialização é um dos comportamentos coletivos mais marcantes observados num sistema complexo de coordenação. Grosso modo, especialização é a divisão do trabalho, alocando agentes específicos para a realização de tarefas distintas. A divisão do trabalho é fundamental para melhorar a eficiência, de modo a obter soluções ótimas para o sistema.

Em sistemas ecológicos, o sinal mais evidente da divisão do trabalho é a existência de castas. Os indivíduos pertencentes a diferentes castas são, geralmente, designados para a realização de tarefas específicas. Podem ser encontrados diversos trabalhos que estudam a formação da especialização e conseqüente divisão de tarefas, como em (El-Mesallamy *et al.*, 2013) e (Meng *et al.*, 2012).

Nos estudos de coordenação, a especialização tem sido estudada como uma propriedade emergente do sistema, ou seja, uma programação *a priori* das características de cada agente raramente acarreta num comportamento ótimo do coletivo, justamente por conta da dificuldade em se analisar os impactos das inter-relações em ambientes complexos. Desta forma, uma especialização emergente se dá através da interação dos agentes em resposta a tarefas dinâmicas que requerem níveis de especialização distintos para serem realizadas.

Além desta classificação, a especialização pode ser classificada em comportamental ou morfológica. Na especialização comportamental, não há uma diferença na estrutura dos agentes, sendo a especialização obtida apenas na mudança de comportamento dos mesmos. Exemplos deste tipo de especialização podem ser vistos em (Vakili *et al.*, 2009) e (Arena, Patane & Vitanza, 2012). A figura 2-4 exhibe duas espécies com distintos tipos de especialização.



Figura 2-4: Especialização Comportamental x Especialização Morfológica.

Entretanto, em algumas situações se faz necessária uma mudança na estrutura dos agentes, seja pelas características dos sensores, seja pela informação disponível para cada agente, ou mesmo por uma limitação física. Desta forma, apenas uma mudança comportamental dos agentes não é suficiente para a atuação ótima da equipe, sendo necessária uma mudança morfológica dos agentes. O modelo NCoQ se enquadra nesta categoria, permitindo que os agentes, em caso de necessidade, evoluam suas estruturas de forma independente para se adaptarem melhor às tarefas que lhes competem. A figura 2-5 exhibe os tipos de especialização e destaca a atuação do NCoQ.

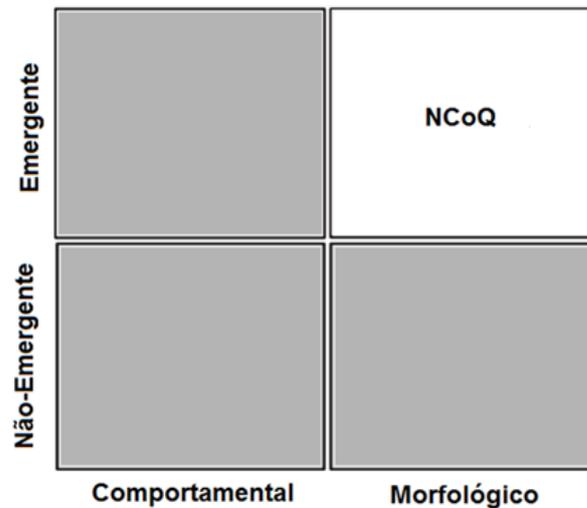


Figura 2-5: Tipos de especialização.

2.1.3.3

Avaliação da Coordenação

Quanto à avaliação da coordenação decorrente de um aprendizado em SMA, de maneira geral, podem-se citar dois tipos (Godoy, 1995):

- Avaliação qualitativa: é o processo que responde à questão “o quão bem nós fazemos?”. O pesquisador não procura enumerar e/ou medir os eventos estudados, nem emprega instrumental estatístico para a análise dos dados, uma vez que é uma avaliação de cunho exploratório, descritivo e subjetivo. Baseia-se em critérios qualitativos, que identificam características adequadas a um problema específico.
- Avaliação quantitativa: é o processo que responde à questão “O quanto nós fazemos?”. O pesquisador preocupa-se com a medição objetiva e a quantificação dos resultados, uma vez que é uma avaliação que se baseia em critérios quantitativos, exibindo resultados numéricos para comparação.

As avaliações são realizadas com os seguintes objetivos (Werkema, 1995):

- Fornecer informação para auxiliar a tomada de decisão na solução de um problema em particular;

- Ajudar a melhorar sistemas, por meio da utilização de técnicas mais adequadas em função das características do problema a ser solucionado;
- Aprender sobre as razões de sucessos e falhas;
- Verificar se o objeto de avaliação atende de forma confiável, acessível, segura e no tempo às necessidades da situação em questão.

Uma das metodologias mais utilizadas para obter a coordenação em SMA é a co-evolução, a ser detalhada no próximo item.

2.2

Neuro-Evolução e Co-Evolução Cooperativa

A neuro-evolução é um modelo híbrido que explora a potencialidade de duas diferentes áreas inspiradas em processos biológicos: Redes Neurais Artificiais (RNA) e Algoritmos Genéticos (AG). A ideia básica dessa metodologia é buscar automaticamente a melhor configuração para uma rede neural usando algoritmos genéticos. Em outras palavras, a neuro-evolução combina a capacidade de generalização e aproximação de funções das redes neurais artificiais com um método eficiente de busca paralela. O objetivo dos algoritmos genéticos é melhorar os algoritmos de aprendizado, automatizando, total ou parcialmente, o processo de configuração da rede neural, bem como o processo de treinamento e atualização dos pesos da mesma.

Co-evolução cooperativa é um método de otimização inspirado na teoria da co-evolução natural que "... ocorre quando duas ou mais espécies influenciam as evoluções umas das outras." (Ridley, 2004). Neste modelo, a aptidão de cada indivíduo não é mais um valor absoluto dado por uma função de avaliação, mas sim um valor relativo baseado em seu desempenho no contexto dos demais indivíduos no mesmo processo de otimização.

A seguir serão apresentados os fundamentos e a revisão de literatura para estas metodologias.

2.2.1

Neuro-Evolução

A fim de aprender um determinado comportamento coletivo, é necessário que a representação de cada agente garanta um meio de traduzir as complexas políticas de controle. As redes neurais artificiais são, em teoria, capazes de aproximar qualquer comportamento com precisão arbitrária se os pesos e topologia estiverem corretamente definidos (Haykin, 2004). Além disso, algoritmos genéticos, que se utilizam dos pressupostos da teoria evolutiva de Darwin para desenvolver soluções, provou ser um método útil para aprender a configurar os pesos e topologias das redes neurais, a fim de resolver diversos problemas (Floreano & Urzelai, 2000), (Stanley, 2004); (Miikkulainen, *et al.*, 2012). Dá-se o nome de neuro-evolução quando algoritmos evolutivos são utilizados para o ajuste de redes neurais. A figura 2-6 ilustra como é feita a evolução das redes neurais.

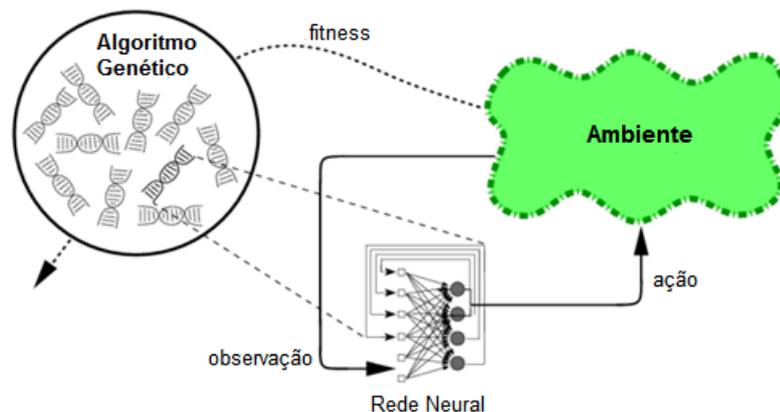


Figura 2-6: Dinâmica da neuro-evolução.

Há outras vantagens no uso da neuro-evolução. (Moriarty, 1997) mostra que métodos evolucionários têm vantagens sobre os métodos tradicionais de aprendizado por reforço por terem um mecanismo de atribuição de crédito mais robusto. (Belew, 1993) afirma que a neuro-evolução, ou evolução simulada de redes neurais, é uma estratégia efetiva para resolver problemas de decisão sequencial, mesmo quando o sinal de reforço é esparsos, ou seja, mesmo quando o agente só é avaliado após a execução de uma sequência de ações e não para cada ação tomada individualmente. (Cruz, 2007) também destaca que não é necessário que o sistema satisfaça nenhuma restrição em particular para que a neuro-evolução funcione. A única exigência é que os mesmos possam, de

alguma forma, ter as suas soluções candidatas avaliadas relativamente umas às outras. Ou seja, o problema deve ser da forma

$$f: S \rightarrow \mathfrak{R} \quad (2.1)$$

onde S é o espaço de busca constituído por todas as possíveis soluções para um problema particular.

É importante mencionar que estas soluções não precisam ser necessariamente ótimas, mas sim satisfatórias. Ao se lidar com espaços de busca grandes e complexos, atingir o ótimo é praticamente impossível e, neste caso, pode-se apenas esperar achar uma solução satisfatória (Zebulum, Pacheco & Vellasco, 2001).

As abordagens possíveis para os sistemas neuro-evolutivos diferem uma das outras, basicamente, pelo modo como as mesmas codificam os pesos e a topologia das redes neurais nos cromossomos. Os cromossomos podem codificar qualquer informação relevante para a parametrização da rede neural, incluindo os pesos sinápticos, o número de camadas escondidas, a topologia da rede, a taxa de aprendizado, etc. A escolha do esquema de codificação tem um papel significativo na formação do espaço de buscas, no comportamento do algoritmo de busca e em como os genótipos devem ser transformados nos fenótipos (representação direta ou indireta).

A seguir, algumas abordagens neuro-evolutivas que foram referências para a criação do modelo NCoQ são apresentadas. Cabe destacar como características comuns ao modelo proposto a evolução concomitante de subpopulações e a capacidade de modificação da arquitetura das redes neurais.

2.2.1.1

Enforced Subpopulations Method (ESP)

A neuro-evolução ESP (*Enforced Subpopulations Method*) (Gomez & Miikkulainen, 1997) é uma extensão do modelo de neuro-evolução SANE (*Symbiotic Adaptive Neuro-Evolution*), (Moriarty & Miikkulainen, 1996). A neuro-evolução SANE se diferencia dos métodos de neuro-evolução tradicionais, pois evolui populações de neurônios, ao invés de redes neurais completas. Cada neurônio da camada escondida contém em sua representação os pesos sinápticos das conexões com os neurônios de saída e com as entradas da rede. Estes neurônios são selecionados aleatoriamente e combinados com o objetivo

de montar uma rede *feed-forward* que é avaliada. O desempenho da rede (*fitness*) é repassado a todos os neurônios que a compõem e, ao final das avaliações, cada neurônio terá uma avaliação média. Então, após as operações de recombinação e mutação, uma nova geração de neurônios é obtida. Ao final de uma evolução SANE, pode-se perceber a formação de *clusters* por parte de alguns neurônios que desenvolveram características semelhantes ao longo do processo.

Na neuro-evolução ESP, a população inicial de neurônios é dividida em subpopulações específicas, de acordo com o número de neurônios na camada escondida da rede, e cada neurônio só pode ser re combinado com neurônios de outras subpopulações. Um esquema pode ser visto na figura 2-7.

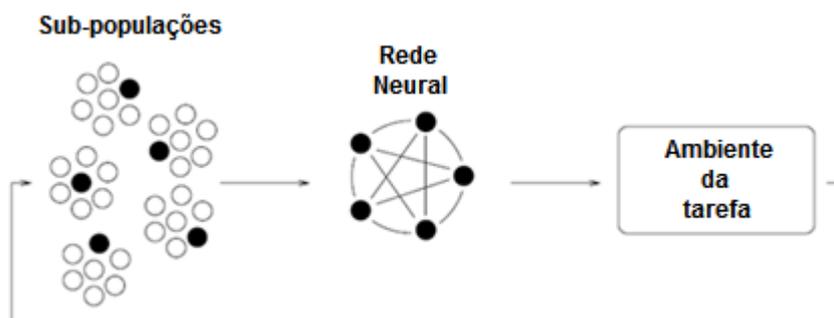


Figura 2-7: Método ESP.

A metodologia ESP apresenta duas vantagens em relação a SANE. Primeiramente, as subpopulações de indivíduos semelhantes que se formam gradualmente em SANE já são inicialmente separadas e isoladas na metodologia ESP. Assim, as “espécies” não têm que se organizar a partir de uma grande população e a evolução das especializações não é prejudicada por cruzamentos entre as espécies. Em segundo lugar, as redes formadas necessariamente contêm um representante de cada subpopulação e o neurônio é avaliado a partir da sua capacidade de adaptação com outros neurônios. Em SANE, as redes podem conter vários membros de uma mesma especialidade ou omitir algumas delas, tornando suas avaliações menos consistentes.

Ainda há a abordagem ESP para problemas Multi-Agente (Gomez & Miikkulainen, 1997). Nesta, cada agente possui as suas subpopulações de neurônios específicas. Este modelo apresentou bons resultados quando comparado a outros modelos neuro-evolutivos como SANE (Moriarty & Miikkulainen, 1996) GENITOR (Whitley *et al.*, 1993), e *Celullar Encoding* (Gruau,

Whitley & Pyeatt, 1996), sendo uma alternativa consistente para a evolução de SMA.

2.2.1.2

Neuro Evolution of Augmenting Topologies (NEAT)

A arquitetura é um parâmetro que interfere diretamente no funcionamento de uma rede neural, sendo sua correta definição de fundamental importância para um bom comportamento do agente. Entretanto, não há na literatura um consenso sobre como fazer tal escolha. Em (Gómez, Franco & Jerez, 2009), por exemplo, podem ser encontradas 4 fórmulas distintas para a determinação do número de neurônios na camada escondida de uma rede *feed-forward*.

Diante deste impasse, é altamente desejável que o algoritmo neuro-evolutivo utilizado seja capaz, não de otimizar apenas os pesos sinápticos que compõem a rede, mas de definir, de maneira autônoma, a arquitetura apropriada para cada situação.

A metodologia NEAT (Stanley, 2004) foi criada com o objetivo de evoluir tanto os pesos sinápticos como a arquitetura da rede. As arquiteturas iniciais são estruturas simples que são incrementadas ao longo da evolução. Para poder realizar cruzamentos entre topologias diferentes, uma nova representação foi criada e pode ser vista na figura 2-8 a seguir.

Cada rede neural é dividida em nós e conexões. Os nós indicam as posições dos neurônios dentro da arquitetura (entrada, saída ou camada escondida). Já as conexões armazenam a ligação entre os nós (entrada e saída), os respectivos pesos e sua habilitação. Além dessas, cada gene adicionado armazena na variável “Inovação” a geração em que foi criado.

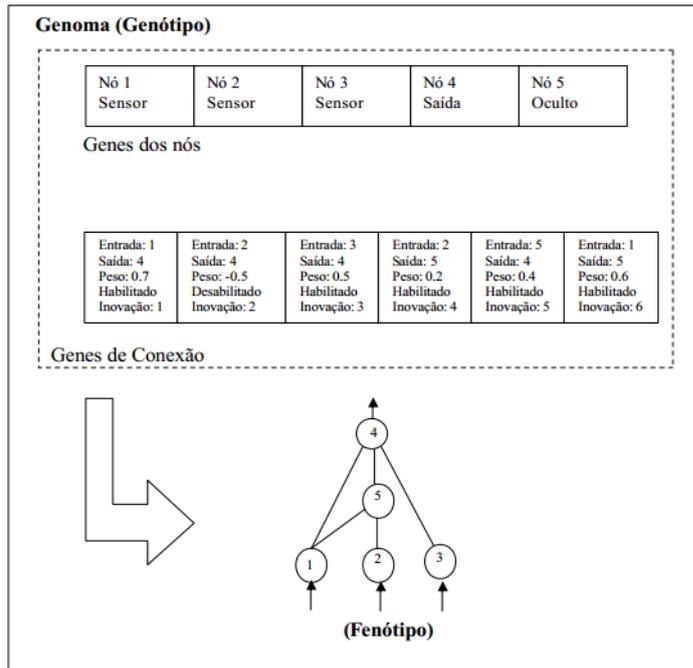


Figura 2-8: Mapeamento do genótipo para o fenótipo.

É a variável “Inovação” que permite o cruzamento entre indivíduos com arquiteturas distintas, conforme ilustra a figura 2-9. Esta figura mostra pais que apresentam estruturas diferentes. Porém, a variável “Inovação” emparelha os genes correspondentes, permitindo a combinação. Os genes do filho são escolhidos aleatoriamente a partir dos pais.

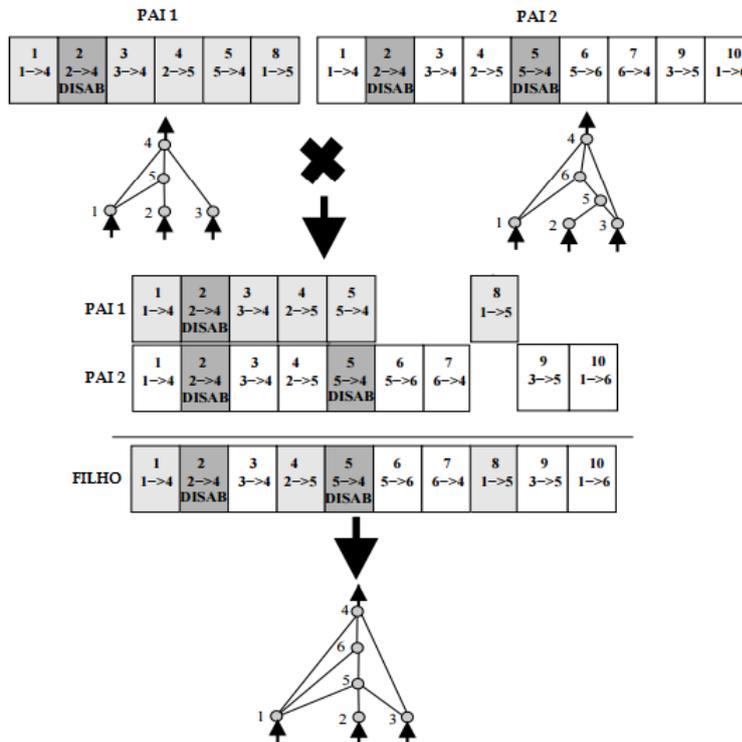


Figura 2-9: Crossover no modelo NEAT.

Na literatura podem ser encontrados diversos trabalhos que utilizam a metodologia NEAT: (Ozveren, Sapeluk & Birch, 2014) utilizaram esta metodologia para problemas de previsão de demanda de energia elétrica no curto prazo; (Tibermacine & Djedi, 2014) criaram e simularam a locomoção de agentes em espaços virtuais; (Sohangir, Rahimi & Gupta, 2013) criaram um modelo NEAT para selecionar variáveis de entrada para o treinamento de redes neurais; no trabalho de (Lowell, Grabkovsky & Birger, 2011) pode-se encontrar uma comparação de desempenho para problemas de tomada de decisão entre o modelo NEAT, com uma variação do mesmo, o Hyper NEAT.

2.2.2

Co-evolução Cooperativa

Algoritmos de co-evolução cooperativa também se utilizam dos pressupostos da teoria evolutiva de Darwin para desenvolver comportamentos coletivos nos agentes. Nesta metodologia, os agentes interagem no mesmo ambiente e evoluem concomitantemente. Dessa forma, o esforço computacional é distribuído sobre a evolução de subpopulações menores, que são agregadas para a montagem de uma solução global.

(Potter & De Jong, 2000) afirmam que para se aplicar algoritmos evolucionários com sucesso em problemas com complexidade cada vez maior, torna-se necessário introduzir noções explícitas de modularidade nas soluções para que elas disponham de oportunidades razoáveis de evoluir na forma de subcomponentes coadaptados. Esta característica pode ser vista em diversos problemas de SMA, geralmente complexos e que podem ser decompostos em subproblemas menores e interdependentes. Além disso, para problemas de coordenação de agentes, a co-evolução se faz necessária pela necessidade de avaliar o indivíduo dentro de um contexto de equipe, ou seja, a aptidão de cada indivíduo não é mais um valor absoluto dado por uma função de avaliação, mas sim um valor relativo baseado em seu desempenho no contexto dos demais indivíduos no mesmo processo de otimização.

Num algoritmo de co-evolução cooperativa, vários algoritmos evolucionários ocorrem em paralelo, cada um responsável por evoluir uma espécie diferente, conforme a figura 2-10, a seguir. Para a avaliação de um indivíduo da espécie 1, por exemplo, é selecionado um indivíduo de cada espécie restante para a montagem de uma equipe que terá a sua capacidade de

coordenação avaliada após a interação com o ambiente. O crédito dessa avaliação retorna ao indivíduo inicial e a todos os outros que participaram da equipe, sendo utilizado para a evolução das espécies pelos algoritmos evolucionários locais.

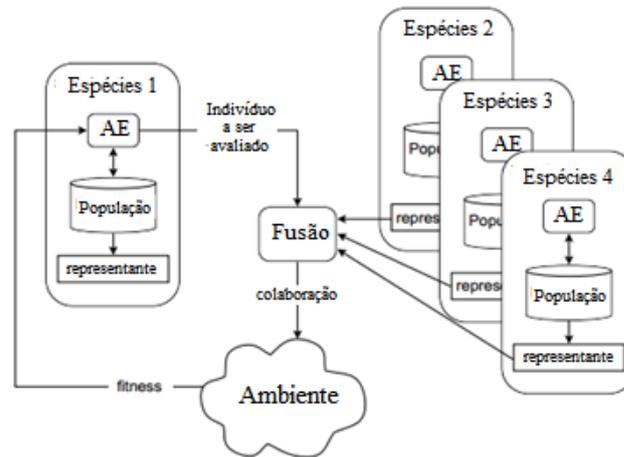


Figura 2-10: Co-evolução cooperativa.

Estudos sobre co-evolução cooperativa podem ser encontrados em (Tonda, Lutton & Squillero, 2012), (Chandra, 2014), (Wei, Wang & Huo, 2013).

No item a seguir, serão apresentados os algoritmos com inspiração quântica que basearam o desenvolvimento do modelo NCoQ.

2.3

Algoritmos Genéticos com Inspiração Quântica

Desde que foram propostos como método de otimização, os algoritmos evolutivos têm sido usados com sucesso para resolver problemas complexos nas mais diversas áreas como, por exemplo, classificação (Barros *et al.*, 2012), *web mining*, (More & Bharambe, 2014), agrupamento (Piquer *et al.*, 2014) e reconstrução de imagens (Rego *et al.*, 2006).

Os algoritmos genéticos com inspiração quântica também fazem uso do conceito de superposição de estados na representação dos genes de seus indivíduos, na tentativa de obter ganhos de convergência durante o processo evolutivo. Neste sentido, representam um dos mais recentes avanços na área de computação evolutiva

Algoritmo evolutivo com inspiração quântica usando representação binária foi proposto inicialmente em (Han & Kim, 2000). Neste modelo, o algoritmo usa

uma representação especial que simula um cromossomo composto por q -bits, onde cada q -bit é formado por um par de números (α, β) , onde $|\alpha|^2 + |\beta|^2 = 1$. O valor dado por $|\alpha|^2$ indica a probabilidade do respectivo q -bit assumir o valor 0 quando for observado, enquanto que a probabilidade de o mesmo assumir valor 1 é dada por $|\beta|^2$. Pode-se visualizar graficamente esta relação entre α e β na figura 2-11 a seguir.

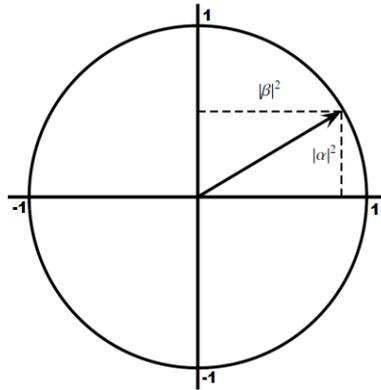


Figura 2-11: Representação gráfica das probabilidades de se observar os valores 0 e 1 para um q -bit qualquer.

Este algoritmo com representação binária foi usado com sucesso em problemas de otimização combinatória (Han & Kim, 2000) e detecção de faces (Jang, Han & Kim, 2004), apresentando resultados superiores aos algoritmos genéticos convencionais em termos de convergência e qualidade de soluções encontradas.

Para a atualização da população quântica são utilizados operadores denominados q -gate. Estes são responsáveis por rotacionar os vetores que representam os bits quânticos, conforme a figura 2-11 em busca de regiões mais promissoras do espaço de busca.

Entretanto, como acontece com outros tipos de algoritmos evolutivos (sendo os algoritmos genéticos o exemplo mais comum), a representação binária não é, necessariamente, a mais adequada para um problema de otimização numérica, por apresentar algumas particularidades que restringem a capacidade de otimização do algoritmo. Em particular, pode-se citar o fato de que a representação binária pode introduzir problemas de métrica relativos à distância entre soluções. Por exemplo, os números 31 e 32 são inteiros próximos, porém são representados na base binária como “01111” e “10000”, ou seja, possuem representações bastante distintas. Isso é um problema do ponto de vista

evolutivo, pois se espera que indivíduos parecidos possuam representações genéticas semelhantes.

2.3.1

AEIQ-R

Por conta disso, optou-se neste trabalho pelo algoritmo evolutivo com inspiração quântica e representação real (AEIQ-R) proposto por (Cruz, 2007) (Cruz & Vellasco, 2010). Neste modelo, os indivíduos da população deixam de ser representados por determinado número de q -bits e passam a ser representados por funções densidade de probabilidade, como na representação a seguir:

$$q_i \rightarrow [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), \dots, g_{iG} = p_{iG}(x)] \quad (2.2)$$

onde $i = 1, 2, 3, \dots, N$, $j = 1, 2, 3, \dots, G$ e as funções p_{ij} representam as funções densidade de probabilidade do gene j do i -ésimo indivíduo quântico.

Esta função densidade de probabilidade é usada pelo AEIQ-R para gerar os valores dos genes dos indivíduos clássicos. Em outras palavras, cada função $p_{ij}(x)$ representa a densidade de probabilidade de se observar um determinado valor para o gene quântico quando a superposição do mesmo for colapsada. A função densidade de probabilidade deve respeitar a propriedade de normalização, ou seja, possuir a integral em seu domínio igual a 1. Ressalta-se que esta adaptação é a mais indicada para a modelagem neuro-evolutiva, por conta da representação contínua do espaço de busca.

Uma das funções mais simples que se pode usar como função densidade de probabilidade é o pulso quadrado. Esta função é definida pela equação:

$$p_{ij}(x) = \begin{cases} \frac{1}{U_{ij} - L_{ij}}, & \text{se } x \in [L_{ij}, U_{ij}] \\ 0, & \text{caso contrário} \end{cases} \quad (2.3)$$

onde L_{ij} é o limite inferior e U_{ij} o limite superior do intervalo no qual o gene j do i -ésimo indivíduo quântico pode assumir valores (colapsar) quando observado. Esta equação respeita a propriedade de normalização mencionada no parágrafo anterior. Além disso, corresponde a uma distribuição de probabilidade que pode ser facilmente utilizada em algoritmos computacionais, já que a mesma

corresponde a uma distribuição uniforme no intervalo $[L_{ij}, U_{ij}]$. A figura 2-12 exhibe a representação do gene quântico feita pelo algoritmo AEIQ-R.

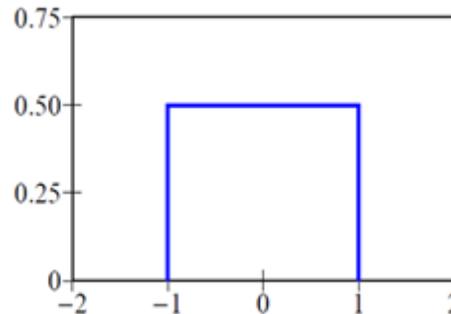


Figura 2-12: Representação do gene quântico.

Um gene quântico pode colapsar mais de uma vez, gerando assim indivíduos clássicos distintos. Na modelagem, tal feito é possível obtendo novos valores aleatórios a partir da distribuição de probabilidade dos pulsos. Estes indivíduos são avaliados e selecionados, sendo os melhores utilizados para atualizar a população quântica. Essa atualização, em geral, deve:

- Reduzir o espaço de busca da função que se deseja otimizar. No AEIQ-R isto é feito reduzindo-se o tamanho da região onde a função densidade de probabilidade (genes quânticos) tem probabilidade diferente de 0;
- Mapear as regiões mais promissoras do espaço de busca. Isto deve ser feito aumentando-se a probabilidade em observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem-sucedidos da população clássica.

No algoritmo AEIQ-R, todos os genes partem da mesma distribuição de probabilidade e sofrem a mesma diminuição percentual da largura dos pulsos. A regra utilizada para a atualização da largura dos pulsos é a heurística proposta por (Michalewicz, 1996) denominada “regra do 1/5” e consiste no seguinte: se menos de 20% da população clássica criada na geração atual tiver uma avaliação melhor do que na geração anterior, a largura do gene é reduzida; se esta taxa for maior do que 20%, a largura do gene é aumentada; caso a taxa seja exatamente igual a 20%, nenhuma alteração é feita. Matematicamente, esta operação pode ser representada pela equação (2.4),

$$\sigma_{ij} = \begin{cases} \sigma_{ij} \cdot \delta, & \varphi < 1/5 \\ \sigma_{ij}/\delta, & \varphi > 1/5 \\ \sigma_{ij}, & \varphi = 1/5 \end{cases} \quad (2.4)$$

onde σ_{ij} é a largura do j -ésimo gene do i -ésimo indivíduo quântico em $Q(t)$, δ é um valor arbitrário, em geral no intervalo $]0,1[$ e φ é a taxa que indica quantos indivíduos da nova população clássica foram melhores do que os indivíduos da população clássica na geração anterior.

Após a atualização da largura dos pulsos, deve-se deslocar o mesmo na direção de indivíduos clássicos que serviram de referência. A figura 2-13 ilustra o processo em que os indivíduos quânticos são colapsados, ou seja, se transformam em indivíduos clássicos que podem ser avaliados e que, depois de selecionados, são utilizados para atualizar a população quântica.

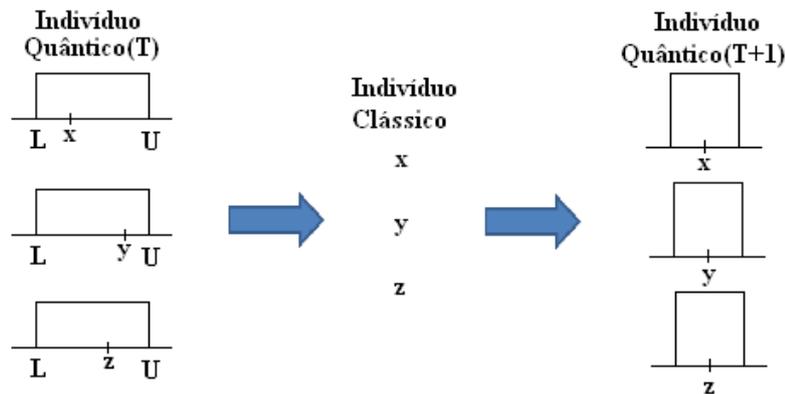


Figura 2-13: Atualização da População Quântica.

O pseudo-código utilizado para a modelagem quântica é apresentado na figura 2-14.

```

iniciar
1.  $t \leftarrow 1$ 
2. Gerar população quântica  $Q(t)$  com  $N$  indivíduos com  $G$  genes
3. enquanto ( $t \leq T$ )
4.      $E(t) \leftarrow$  gerar indivíduos clássicos usando indivíduos quânticos
5.     se ( $t = 1$ ) então
6.          $C(t) \leftarrow E(t)$ 
7.     senão
8.          $E(t) \leftarrow$  recombinação entre  $E(t)$  e  $C(t)$ 
9.         Avaliar  $E(t)$ 
10.         $C(t) \leftarrow K$  melhores indivíduos de  $[E(t)+C(t)]$ 
11.    fim se
12.     $Q(t+1) \leftarrow$  Atualiza  $Q(t)$  usando os  $N$  melhores indivíduos de  $C(t)$ 
13.     $t \leftarrow t + 1$ 
14. fim enquanto
fim

```

Figura 2-14: Listagem completa do algoritmo evolutivo com inspiração quântica usando representação real.

2.3.2

Avanços na modelagem quântica

Assim como (Cruz, 2007) propôs o modelo AEIQ-R com o objetivo de estender o modelo binário proposto por (Han & Kim, 2004) para problemas de otimização numérica, autores diversos propuseram melhorias e combinações de técnicas utilizando os algoritmos evolutivos com inspiração quântica.

(Pinho, 2010) desenvolveu um modelo híbrido entre o AEIQ-B e AEIQ-R, criando o modelo AEIQ-BR, que utiliza representações binárias para definir os parâmetros da rede neural, como número de neurônios nas camadas escondida e de saída, e a representação real para otimizar os pesos sinápticos da rede. Dois tipos de arquitetura foram testados: redes *feed-forward* para problemas de classificação e recorrentes para problemas de controle, obtendo bons desempenhos.

(Dias, 2010) apresentou um novo modelo de Programação Genética Linear com Inspiração Quântica (PGLIQ) para evoluir programas em código de

máquina. Nos testes comparativos foram utilizados problemas de regressão simbólica e a PGLIQ apresentou desempenho superior, obtendo melhores soluções com um número menor de avaliações, parâmetros e operadores.

Alguns estudos em algoritmos evolutivos quânticos propõem operadores para serem empregados em indivíduos no estado quântico, como *crossover* e *mutação* quânticos sugeridos por (Mohammed *et al.*, 2012). As figuras 2-15 e 2-16 ilustram, respectivamente, os operadores de *crossover One-Point* e *Quantum Inversion Mutation Operator*.

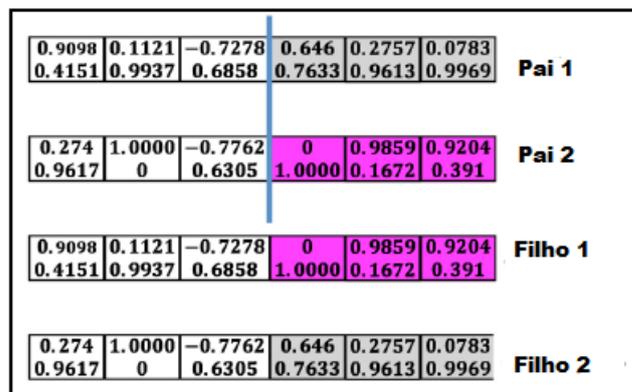


Figura 2-15: Cross Over Quântico - One Point.

Na figura 2-15 é destacado o ponto de corte para a realização do crossover em dois indivíduos com 6 bits quânticos cada, gerando 2 filhos com uma parte de cada pai.

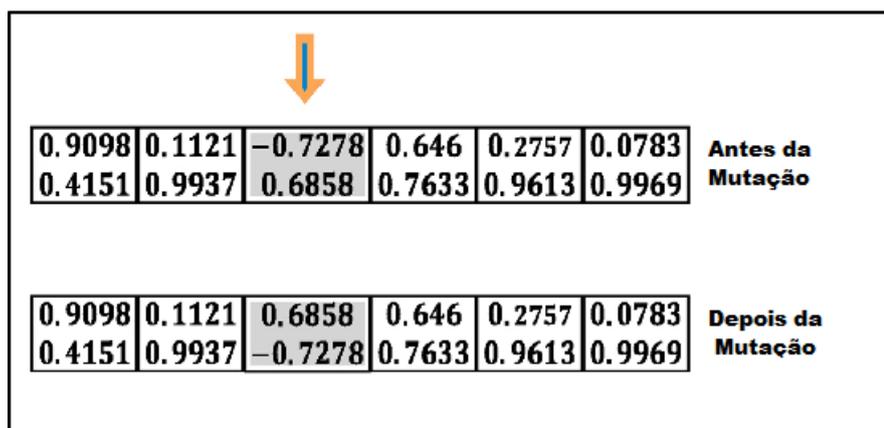


Figura 2-16: Quantum Inversion Mutation Operator.

Na figura 2-16, o gene escolhido para a mutação está em destaque. Repare na troca de posições de α e β no q -bit.

Já (Sheng & Wanlu, 2012) propuseram modificações no operador de atualização da população quântica, o *q-gate*, atualizando o ângulo de rotação através do método de enxame de partículas (Kennedy & Eberhart, 1995). Os experimentos mostraram que este tipo de operador aumentou a capacidade de busca global quando comparado a algoritmos genéticos tradicionais.

Em relação aos operadores de mutação, devido à natureza aleatória dos algoritmos com inspiração quântica, recomenda-se a não utilização dos mesmos ou a utilização com baixa probabilidade. (Mohammed *et al.*, 2012) desenvolveram o operador de mutação quântico *swap* que, combinado a outros operadores, obtiveram melhoras de desempenho em torno de 15%. Neste operador, dois genes do indivíduo são escolhidos aleatoriamente para trocarem de posição, como pode ser visto na figura 2-17.

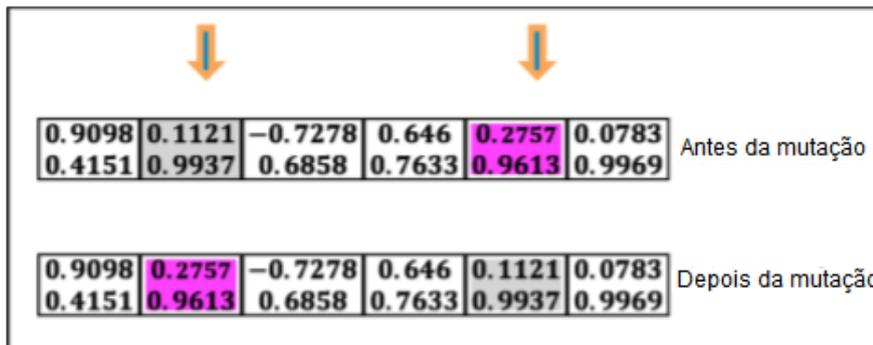


Figura 2-17: Mutação Quântica Swap.

Diante da melhoria obtida pelos novos operadores quânticos para representação binária, surge o questionamento se operadores semelhantes para a representação por pulsos quânticos poderiam acarretar algum ganho de desempenho. Ao final do próximo capítulo, operadores quânticos de *crossover* e mutação para a representação numérica serão propostos e comparados na otimização de funções.

No próximo capítulo, o modelo NCoQ é apresentado, detalhando-se o seu funcionamento.

3

Modelo de Neuro Co-evolução com Inspiração Quântica (NCoQ)

Nos capítulos anteriores, foram abordados os problemas de coordenação nos SMA com suas características e principais metodologias apresentadas, sendo a coordenação entre os agentes um dos grandes desafios para este tipo de problema. Isso porque a coordenação ideal vai depender das características particulares de cada problema e das possibilidades das ações de cada agente. Além disso, o espaço de busca aumenta exponencialmente conforme aumenta o número de agentes a serem coordenados.

O objetivo deste capítulo é apresentar o modelo evolutivo proposto nesta tese para a coordenação de equipes em resolução de tarefas de coordenação. Este modelo deve ser capaz de encontrar soluções razoáveis para qualquer tipo de problema deste tipo, sem a interferência de um programador, exigindo apenas que as soluções criadas ao longo da evolução possam ser avaliadas.

O modelo proposto de Neuro Co-evolução com Inspiração Quântica (NCoQ) tem como base o algoritmo AEIQ-R proposto por (Cruz, 2007), devido à representação real utilizada pelo mesmo, sendo esta a mais indicada para um problema de otimização numérica como a neuro-evolução. Além disso, conforme mencionado anteriormente, a inspiração quântica permite a redução do número de indivíduos dentro das populações, o que representa um grande ganho de processamento em uma co-evolução onde há diversas populações evoluindo concomitantemente.

A co-evolução permite a seleção dos indivíduos que se adaptam melhor ao coletivo, pois o desempenho dos indivíduos é medido avaliando-se o trabalho em equipe. Dessa forma, espera-se que a coordenação seja obtida através da interação entre os agentes, dispensando-se a interferência do programador.

Outra característica importante do modelo NCoQ é a capacidade de evolução de arquitetura das redes neurais responsáveis pelo controle dos agentes. Sendo assim, o programador não precisa se preocupar em determinar de maneira exata o número de neurônios na camada escondida, bastando atribuir um valor inicial para tal parâmetro que, durante a evolução, testes com retiradas e acréscimos de neurônios serão feitos visando a melhoria das soluções. Vale ressaltar que o modelo NCoQ permite arquiteturas distintas entre

os agentes da equipe, o que é um fator altamente desejável para SMA heterogêneos.

Vale ressaltar que não há comunicação explícita entre os agentes, sendo a coordenação obtida através da evolução do comportamento dos mesmos durante os treinamentos, realizados em ambiente pré-definido.

3.1

Modelo NCoQ

O Modelo de Neuro Co-Evolução com Inspiração Quântica (NCoQ) une o potencial da co-evolução em SMA cooperativos com os benefícios do uso dos paradigmas da física quântica em modelos neuro-evolutivos, tendo por base o modelo AEIQ-R proposto por (Cruz, 2007). O modelo AEIQ-R utiliza o conceito de pulsos quânticos para neuro-evoluir uma rede neural recorrente, ao invés de uma representação binária que não é recomendada para esse propósito. Com isso, apenas um pulso quântico consegue representar uma gama infinita de possíveis pesos para a rede, o que diminui a população necessária para a evolução e, conseqüentemente, o esforço computacional.

Basicamente, o modelo NCoQ é composto de 3 etapas, interdependentes, com as seguintes finalidades:

- Co-evolução – parte responsável pela coordenação dos agentes;
- Evolução Quântica – responsável por evoluir cada subpopulação de indivíduos quânticos;
- Neuro-Evolução Clássica – responsável por evoluir as melhores soluções encontradas (redes neurais).

O pseudo-código do algoritmo pode ser visto na figura 3-1 a seguir.

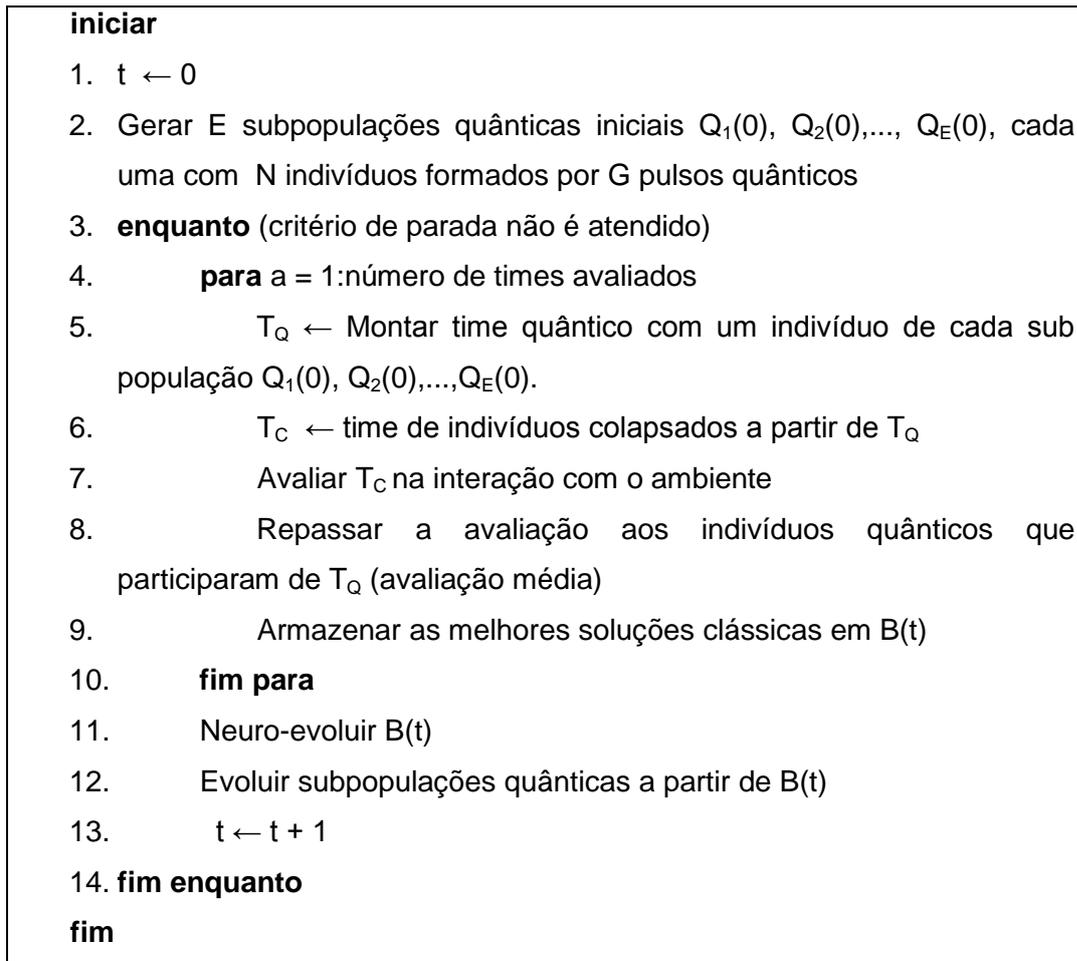


Figura 3-1: Algoritmo de Co-evolução com destaque para os demais blocos de instrução.

A seguir, serão descritos com maiores detalhes os passos mais importantes do algoritmo NCoQ: a geração da população quântica, a avaliação dos times e o repasse da mesma aos indivíduos, bem como as configurações e parâmetros das redes neurais utilizadas no modelo neuro-evolutivo.

3.1.1

População Quântica

O algoritmo NCoQ tem início com a criação das subpopulações quânticas iniciais, sendo necessário para tal a definição dos seguintes parâmetros:

- Limites inferiores e superiores iniciais (L_{ij} , U_{ij}): limites iniciais para os pulsos quânticos (equação 2.3).
- Número de genes necessários para formar cada indivíduo (G): dependerá das dimensões de entrada e saída da rede e do número de neurônios na camada escondida;

- Número de indivíduos das subpopulações (N): quantidade de indivíduos quânticos presentes em cada subpopulação. Esse número permanece fixo durante todo o processo. É recomendável que seja um número pequeno, por conta da característica quântica dos indivíduos.
- Número de especialistas (E): quantidade de agentes que serão coordenados. Equivale ao número de subpopulações a serem evoluídas;

Inicialmente, é gerada uma subpopulação quântica para cada agente. Portanto, em um instante t qualquer, cada subpopulação quântica é formada por um conjunto de N indivíduos quânticos q_i ($i = 1, 2, 3, \dots, N$). Cada indivíduo quântico q_i desta população é formado por G genes g_{ij} ($j = 1, 2, 3, \dots, G$) que, por sua vez, são formados por funções que representam uma densidade de probabilidade. Cada função será armazenada através de dois parâmetros dados pelos limites inferior e superior do pulso (L_{ij}, U_{ij}). Portanto, cada indivíduo quântico segue a equação 2.2. e possui dimensão $(G,2)$.

Após a definição destes parâmetros, é gerada a matriz de dimensão $(E,N,G,2)$ que armazena a população quântica inicial. A figura 3-2 mostra passo-a-passo a composição da população quântica.

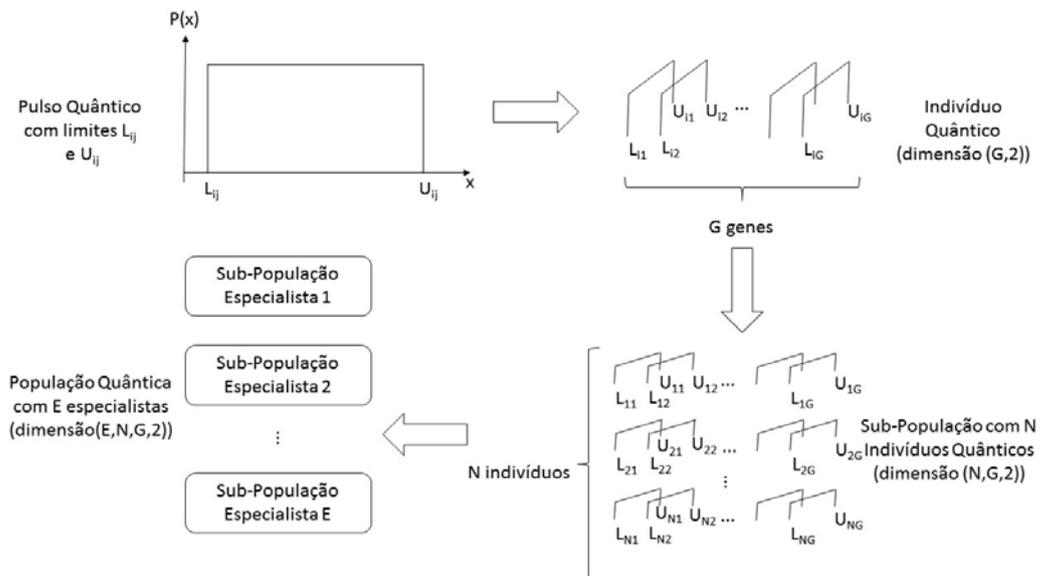


Figura 3-2: População Quântica formada pelas subpopulações de especialistas.

O processo de montagem dos times e de avaliação dos indivíduos será explicado no próximo item.

3.1.2

Time quântico e a avaliação dos indivíduos

A próxima etapa do algoritmo, após a criação das subpopulações, é a avaliação dos indivíduos quânticos levando-se em consideração a capacidade dos mesmos em se organizarem com os indivíduos das outras subpopulações. Para tal, os indivíduos quânticos são organizados em times quânticos. Um time quântico é formado pela escolha aleatória de um único indivíduo de cada subpopulação quântica.

Para a avaliação de um time quântico é necessário que o mesmo seja colapsado, pois apenas indivíduos clássicos conseguem ser avaliados diretamente. Uma propriedade interessante dos algoritmos com inspiração quântica é que um indivíduo quântico pode sofrer vários colapsos e dar origem a indivíduos clássicos diferentes. Neste caso, o número de colapsos também é um parâmetro. Entretanto, ressalta-se que o número de colapsos deve ser pequeno de forma a diminuir o número de avaliações, facilitando assim a convergência do algoritmo.

Após colapsados e avaliados todos os times, cada indivíduo quântico recebe como avaliação a média das avaliações dos times clássicos colapsados em que teve participação (linha 8, figura 3-1). Dessa forma, os indivíduos quânticos são avaliados pelo desempenho dos indivíduos clássicos gerados. Esta característica é uma inovação em relação ao modelo AEIQ-R. A figura 3-3 a seguir ilustra a dinâmica para a avaliação de um time quântico com dois agentes colapsando 3 vezes.

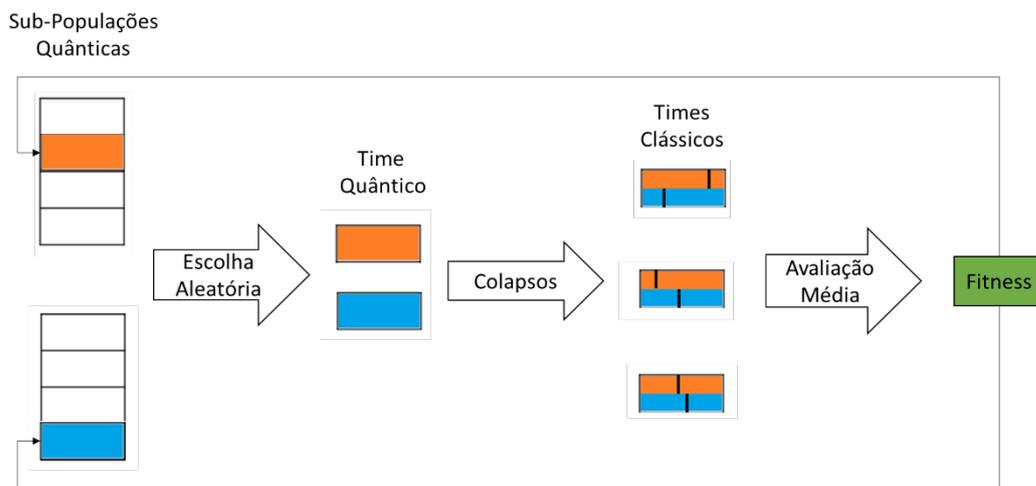


Figura 3-3: Dinâmica de avaliação dos indivíduos quânticos.

Um dos parâmetros a ser definido neste processo co-evolutivo é o número de avaliações a serem feitas para cada indivíduo da população. (Gomez & Miikkulainen, 1997) sugerem a quantidade de 10 avaliações, em média, para cada indivíduo. Por exemplo, se o número de indivíduos em cada subpopulação for 8, serão montados 80 times para avaliação. Admitindo que existam 3 especialidades e, portanto, 3 subpopulações, o total de times que podem ser formados é de $8^3 = 512$. Note que no exemplo citado, 15,63% das combinações possíveis de equipes serão avaliadas. Porém, os indivíduos, em média, serão avaliados 10 vezes.

Cabe ressaltar que, embora o algoritmo possua inspiração quântica, as soluções obtidas devem ser clássicas com a finalidade de serem utilizadas nas aplicações. Portanto, os indivíduos clássicos que foram melhor avaliados são armazenados (linha 9, figura 3-1) para serem aproveitados na etapa de neuro-evolução.

Ao final de cada geração t tem-se duas populações de naturezas distintas:

- Um conjunto de E subpopulações $Q_1(t), Q_2(t), \dots, Q_E(t)$ de indivíduos quânticos com suas respectivas avaliações;
- Um conjunto $B(t)$ das equipes clássicas melhor avaliadas e que será neuro-evoluído em busca de soluções mais refinadas, de acordo com o item a seguir.

3.1.2

Neuro-evolução clássica e o operador *lesion* multi-agente

Por conta da natureza aleatória e de busca paralela envolvida nos algoritmos quânticos, percebeu-se a necessidade de um mecanismo que permitisse uma busca local (*exploitation*), refinando as soluções encontradas.

Este processo, ao selecionar a melhor solução clássica, também afetará:

- a atualização dos indivíduos quânticos: os operadores quânticos tendem a deslocar o centro do pulso na direção da melhor solução;
- a arquitetura da rede: a estabilização da melhor solução será o gatilho para o operador *lesion*, e a melhor configuração encontrada será repassada aos demais indivíduos.

Para a arquitetura da rede, optou-se por uma rede neural recorrente, semelhante à apresentada pela figura 3-4, onde podem-se variar as dimensões de entrada e saída, bem como o número de processadores da rede neural.

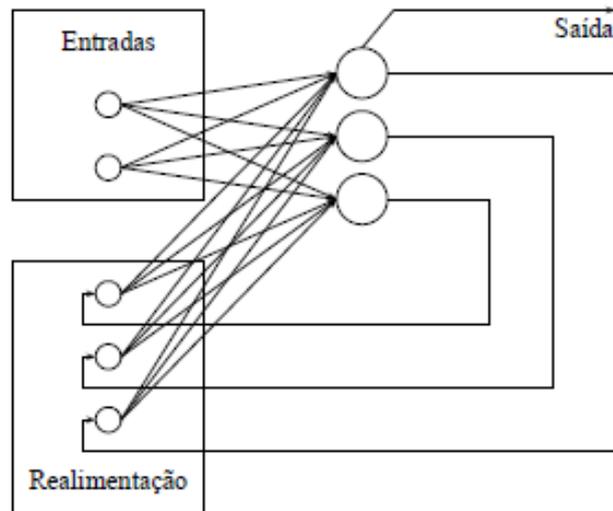


Figura 3-4: Arquitetura da rede neural recorrente.

A escolha por esse tipo de arquitetura é justificada pelos seguintes motivos:

- Redes recorrentes possuem bom desempenho em problemas de previsão de séries (Haykin, 2004) e problemas de controle (Gomez & Miikkulainen, 2003), pelo fato das retroalimentações introduzirem efeitos de memória;
- (Gomez & Miikkulainen, 1997) recomendam esta arquitetura para a neuro-evolução ESP.
- É uma arquitetura mais generalizada, pois se os pesos de realimentação forem zeros, a rede recorrente se transforma em uma rede *feed-forward*; entretanto, a transformação contrária não é possível.

Ressalta-se que os parâmetros de entrada e saída dependerão exclusivamente do problema a ser abordado. Entretanto, apesar de inicialmente determinado, o número de neurônios na camada escondida pode variar, conforme veremos adiante.

Sendo assim, admitindo a entrada da rede com dimensão “e”, a saída com dimensão “s” e “r” neurônios na camada recorrente, o número de genes G na representação genética de um indivíduo clássico será de $G = r.e + r^2 + r$. A primeira parcela se refere ao número de pesos sinápticos necessários entre a

entrada e a camada recorrente; a parcela intermediária corresponde aos pesos de retroalimentação; e a última parcela corresponde aos *bias* de cada processador. A única restrição para esses parâmetros é, obviamente, que a dimensão da saída não supere a quantidade de processadores.

Ressalta-se que o número de genes para os indivíduos quânticos seguirá o número de genes da população clássica corrente.

Na figura 3-5 a seguir, pode-se observar a representação de um indivíduo, reorganizado na forma matricial para ser usado na rede recorrente.

E_{11}	E_{12}	...	E_{1e}	R_{11}	R_{12}	R_{1r}	B_1
E_{21}	E_{22}	...	E_{2e}	R_{21}	R_{22}	R_{2r}	B_2
...							
E_{r1}	E_{r2}	...	E_{re}	R_{r1}	R_{r2}	R_{rr}	B_r

Figura 3-5: Representação matricial do indivíduo utilizado.

Note que ao aumentar um neurônio na camada de retroalimentação, a representação dos indivíduos sofre um aumento de $e + r + 1$ genes. Isso significa que, se a dimensão da entrada for 6 e o número de processadores for 10, havendo a possibilidade de retirada de um neurônio sem prejuízo significativo para o desempenho da rede, haveria uma redução de 17 parâmetros a serem otimizados. Essa economia aumenta o desempenho da evolução genética e antecipa a convergência.

Entretanto, o número de neurônios existentes, bem como a arquitetura são parâmetros fundamentais para o desempenho de uma rede neural. Processadores em excesso podem dar à rede a capacidade de “memorizar” o conjunto de treinamento, diminuindo a capacidade de generalização do sistema. Em contrapartida, poucos neurônios podem inviabilizar o processo de aprendizado ou retardá-lo (Haykin, 2004).

Visando minimizar a interferência do programador e dar maior autonomia ao aprendizado, optou-se por, assim como em (Cruz, 2007), utilizar o operador *lesion* proposto por (Gomez & Miikkulainen, 2003). Quando a evolução da melhor solução obtida se mantém constante por um determinado número de gerações η (especificado pelo usuário), retira-se um neurônio da camada escondida e faz-se uma nova avaliação da rede. Se a avaliação não diminuir até um determinado nível, presume-se que este neurônio não é importante e o

mesmo é retirado da rede. Este processo é repetido para cada neurônio da camada escondida. Caso contrário, ou seja, se o desempenho da rede apresenta piora significativa com a retirada de qualquer neurônio, é possível que a mesma precise de mais um neurônio. Então, acrescenta-se um neurônio com os respectivos pesos sinápticos zerados, com o objetivo de não alterar a avaliação inicialmente.

Ressalta-se que a representação utilizada neste trabalho não possui a capacidade de realizar o cruzamento entre indivíduos de arquiteturas distintas. Sendo assim, a nova arquitetura obtida através da atuação do operador *lesion* é replicada a todas subpopulações de agentes. Ou seja, se o operador define que determinado neurônio deve ser retirado de um agente, todos os indivíduos da respectiva subpopulação deste agente terão o mesmo neurônio retirado. Da mesma forma, se o operador identifica a necessidade de acrescentar um neurônio em determinado agente, esse acréscimo é feito naquela subpopulação correspondente.

Portanto, espera-se que o operador *lesion* diminua o esforço computacional exigido para a identificação do número ideal de processadores e da arquitetura a ser utilizada.

3.2

Evolução Quântica

Neste item, será descrito em maiores detalhes como se dá o processo de evolução da população quântica. Assim, como no AEIQ-R, a atualização da população quântica tem por objetivo:

- Reduzir o espaço de busca da função que se quer otimizar. Isto é feito reduzindo-se o tamanho da região onde a função densidade de probabilidade (genes quânticos) tem probabilidade diferente de 0;
- Mapear as regiões mais promissoras do espaço de busca. Isto deve ser feito aumentando a probabilidade de se observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem sucedidos da população clássica.

São basicamente duas modificações a serem feitas nas subpopulações quânticas a serem atualizadas:

- 1) Modificar a largura dos pulsos de modo que o espaço de buscas seja reduzido;
- 2) Modificar a posição dos pulsos de modo que o ponto central dos mesmos coincida com o valor dos genes de um conjunto de indivíduos da população.

Aqui cabe uma ressalva importante: uma das diferenças entre o algoritmo AEIQ-R (Cruz, 2007) e o algoritmo NCoQ é a capacidade do segundo em avaliar os indivíduos quânticos a partir do desempenho médio das equipes. Desta forma, surge uma alternativa à regra do 1/5 para a atualização da população quântica, mais próxima da dinâmica utilizada em algoritmos genéticos tradicionais, através da aplicação dos operadores de mutação e *crossover*.

Uma das possibilidades para a geração de novos indivíduos quânticos pode ser vista na figura 3-6.

Durante o processo evolutivo, os operadores de cruzamento e mutação exercem papel crucial na capacidade de busca e convergência do algoritmo. Estes operadores são necessários por gerar diversidades na população, mantendo as características adaptativas das gerações anteriores. Os parâmetros utilizados por estes operadores, bem como o tipo de operador utilizado, afetam diretamente o desempenho da neuro-evolução. Os efeitos decorrentes da escolha inadequada destes parâmetros vão desde aumento no tempo de convergência, convergência prematura, estagnação da busca e maior necessidade de recursos computacionais, até a não convergência para uma solução viável.

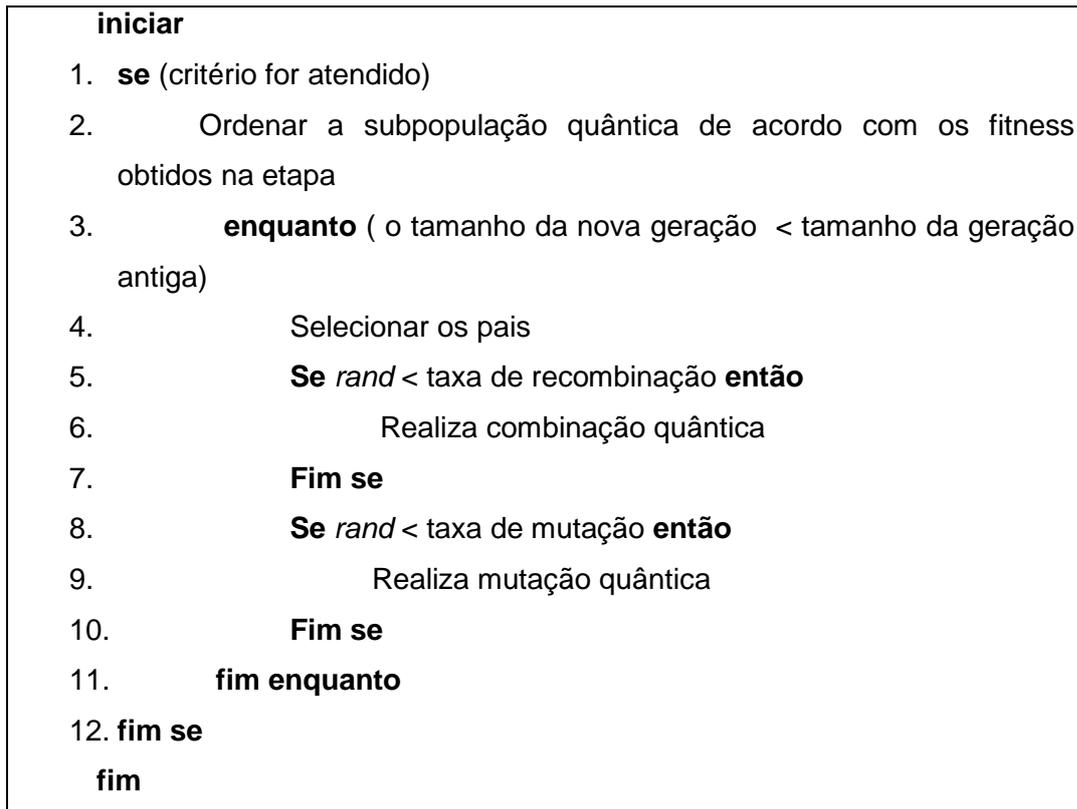


Figura 3-6: Evolução da população quântica.

Neste sentido, o desenvolvimento de operadores mais eficientes tem como consequência um melhor desempenho no processo de evolução, assim como os operadores quânticos desenvolvidos por (Mohammed *et al.*, 2012) e exibidos na seção 0., que apresentaram ganhos significativos de desempenho, próximos de 80% em alguns casos estudados.

3.2.1

Desenvolvimento e testes de operadores quânticos

Na literatura, já podem ser encontrados operadores genéticos de natureza quântica, mas estes não lidam com a representação real através de pulsos quânticos.

Neste sentido, foi desenvolvido o operador de *crossover* aritmético para pulsos quânticos, com a média e a largura dos filhos gerados em função dos pulsos dos pais, dados pelas equações 3.1 e 3.2 a seguir:

Crossover aritmético quântico

$$\mu_{\text{filho}} = \left(\frac{1}{L_{\text{pai 1}}} \mu_{\text{pai 1}} + \frac{1}{L_{\text{pai 2}}} \mu_{\text{pai 2}} \right) \left(\frac{1}{L_{\text{pai 1}}} + \frac{1}{L_{\text{pai 2}}} \right) \quad (3.1)$$

$$L_{\text{filho}} = \frac{L_{\text{pai 1}} + L_{\text{pai 2}}}{2} \quad (3.2)$$

onde μ representa a média e L a largura dos pulsos.

A equação 3.1 leva em consideração que a representatividade da média de uma variável aleatória qualquer é inversamente proporcional à largura dos pulsos. Ou seja, se dentre os pais selecionados houver algum onde a largura do pulso é pequena, aumentando-se assim a probabilidade em torno da média, a média deste terá um peso maior para a geração da média do filho. Ou seja, a média do filho é obtida através de uma média ponderada dos pais, com os pesos inversamente proporcionais a suas respectivas larguras de pulso.

Já a equação 3.2, que mede a largura do novo pulso, visa não diminuir bruscamente o espaço de busca, o que poderia causar uma convergência prematura, nem aumentá-lo demasiadamente, em contradição ao objetivo de atualização da população quântica. Logo, a média aritmética entre as larguras dos pulsos é o mais recomendado.

Por exemplo, considere que dois pais foram selecionados, sendo um com média 8 e largura de pulso 2 e outro com média 0 e largura de pulso 10. Tendo-se em vista que média do primeiro pai é muito mais representativa, já que a largura do pulso do mesmo é 5 vezes menor, a média do filho se aproximará mais da média deste. Aplicando as equações, o seguinte filho será gerado:

$$\mu_{\text{filho}} = \left(\frac{1}{2} \times 8 + \frac{1}{10} \times 0 \right) \left(\frac{1}{2} + \frac{1}{10} \right) = 6,67 \quad (3.3)$$

$$L_{\text{filho}} = \frac{2+10}{2} = 6 \quad (3.4)$$

Uma visualização gráfica do exemplo acima pode ser vista na figura 3-7 a seguir, onde os pais são representados pelos pulsos azuis e vermelhos e o filho pelo pulso roxo.

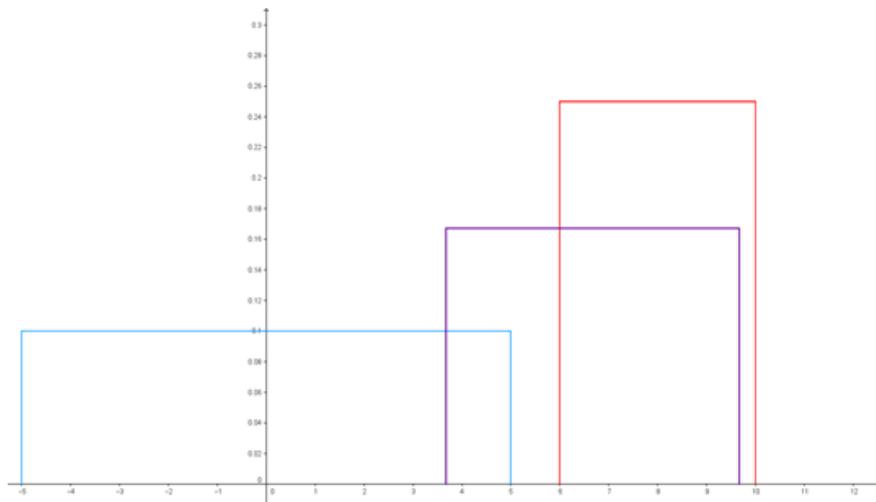


Figura 3-7: Visualização do operador de *crossover* aritmético quântico proposto.

O operador de mutação quântica, proposto nesta tese, busca modificações com o objetivo de evitar a estagnação no espaço de busca. Como os algoritmos com inspiração quântica possuem uma característica aleatória evidente, alguns autores recomendam a omissão desse operador no processo. Entretanto, a mutação aqui proposta tem por objetivo deslocar os pulsos selecionados em direção à melhor solução vigente e diminuir os espaços de busca. As equações da mutação para pulsos quânticos podem ser vistas a seguir.

Mutação aritmética para pulsos quânticos

$$\mu_{\text{filho}} = \mu_{\text{pai}} + (\mu_{\text{best}} - \mu_{\text{pai}}) \cdot \Delta x, \quad 0 < \Delta x < 1 \quad (3.5)$$

$$L_{\text{filho}} = L_{\text{pai}} \cdot \Delta x, \quad 0 < \Delta x < 1 \quad (3.6)$$

Admita, por exemplo, que o pulso a ser modificado tenha média 3 e largura 4, e que o valor de referência para este gene, encontrado na melhor avaliação corrente, é de -1. Se o Δx aleatório for igual 0,6, a média e a largura do novo indivíduo serão dados por:

$$\mu_{\text{filho}} = 3 + (-1 - 3) \cdot 0,6 = 0,6 \quad (3.7)$$

$$L_{\text{filho}} = 4 \cdot 0,6 = 2,4 \quad (3.8)$$

Para um melhor entendimento, a figura 3-8 exhibe graficamente esta operação.

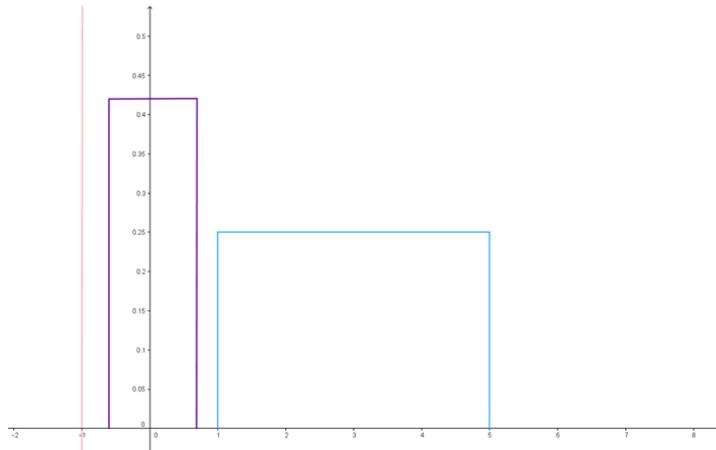


Figura 3-8: Visualização do operador de mutação aritmética quântica proposto.

Os operadores quânticos de crossover *one-point* e *two-points*, bem como o operador quântico *swap* de mutação, propostos por (Mohammed *et al*, 2012) para a representação binária, foram adaptados para poderem ser utilizados na representação numérica.

No item a seguir, serão apresentados os resultados dos testes feitos com diversas combinações destes operadores, a fim de verificar o desempenho dos mesmos.

3.3

Resultados – Otimização de funções e Comparação de Operadores Quânticos

Para a avaliação dos operadores quânticos e comparação dos mesmos com outros operadores genéticos encontrados na literatura, foram utilizadas 4 funções *benchmarks* encontradas em (Jamil & Yang, 2013) e que podem ser vistas na tabela 3-1. O valor ótimo de cada função é representado por $F(x^*)$.

Optou-se por estas funções pelas diferentes dimensões das mesmas (2, 4, 5 e 7 respectivamente), sendo o gráfico da função F1 de duas variáveis podendo ser visto na figura 3-9.

Tabela 3-1: Funções *benchmark* utilizadas.

Função	Equação	Domínio (x_i)	$F(x^*)$
F1 - Ackley 2	$f(x) = -200e^{-0,02\sqrt{x_1^2+x_2^2}}$	$[-32,32]$	-200
F2 - Colville	$f(x) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10,1((x_2 - 1)^2 + (x_4 - 1)^2) + 19,8(x_2 - 1)(x_4 - 1)$	$[-10,10]$	0
F3 - Dolan	$f(x) = (x_1 + 1,7x_2) \sin(x_1) - 1,5x_3 - 0,1x_4 \cos(x_4 + x_5 - x_1) + 0,2x_5^2 - x_2 - 1$	$[-100,100]$	-529,87
F4 - Alpine 2	$f(x) = \prod_{i=1}^7 \sqrt{x_i} \sin(x_i)$	$[0,10]$	1376,51

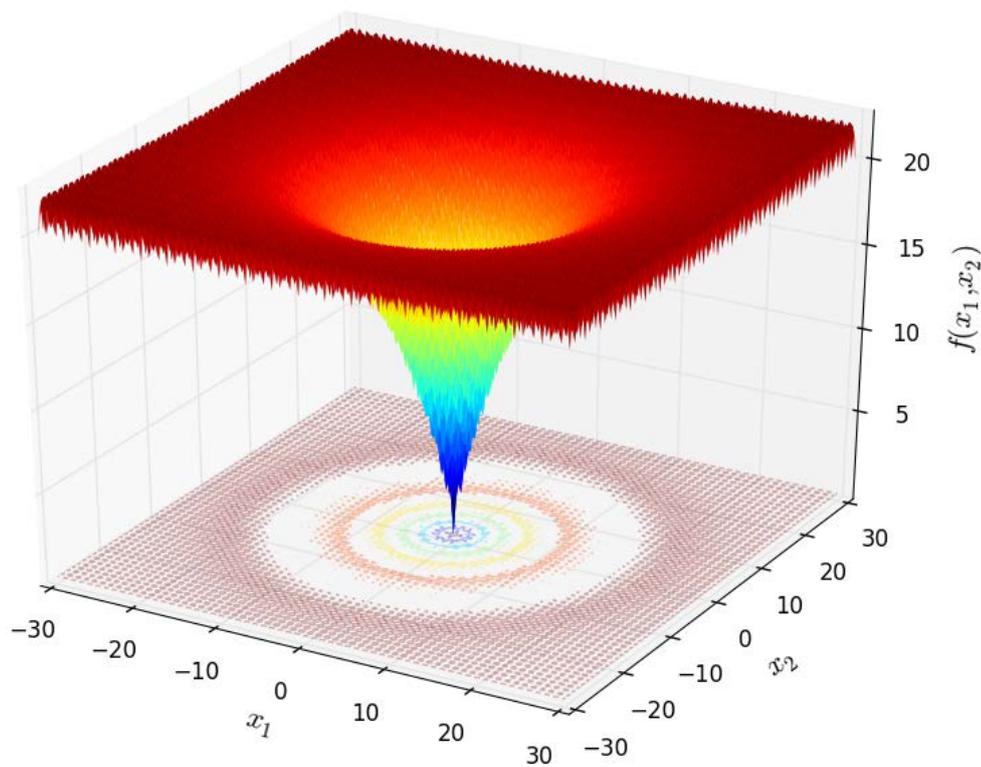


Figura 3-9: Função Ackley 2.

Para a realização dos testes, além da simulação feita utilizando a regra do 1/5, foram realizadas mais 6 simulações, combinando os 3 operadores de *crossover* com os 2 de mutação, seguindo as legendas exibidas na tabela 3-2.

Tabela 3-2: Legendas dos operadores quânticos.

Operadores Quânticos de <i>Crossover</i>		Operadores Quânticos de Mutação	
Legenda	Nome	Legenda	Nome
X1	One-Point	M1	Swap
X2	Two-Point	M2	Aritmético
X3	Aritmético		

Em relação aos parâmetros a serem utilizados, optou-se, visando simplificar, pela manutenção dos mesmos durante os testes. Estes podem ser vistos na tabela 3-3.

Tabela 3-3: Parâmetros utilizados nos testes de otimização.

Parâmetro	Valor
Taxa de <i>crossover</i> quântico	60%
Taxa de mutação quântica	5%
Gerações	200
Número de Indivíduos Quânticos	20
Número de colapsos	3
Taxa de <i>crossover</i> clássico	80%
Taxa de mutação clássica	10%
Elitismo	35%
δ (regra do 1/5)	20%

Os resultados a seguir foram obtidos após 25 rodadas de simulação para cada configuração proposta. Os erros foram calculados a partir do ponto ótimo conhecido de cada função. Além disso, visando comparar os operadores entre si, foram calculados os desvios percentuais relativos (DPR) dado pela fórmula a seguir:

$$DPR = \frac{e_{alg} - e_{min}}{e_{min}}, \quad (3.9)$$

onde e_{alg} é o erro do algoritmo em questão e e_{min} é o erro mínimo encontrado pelos algoritmos propostos, ou seja, é o erro do algoritmo que apresentou o melhor desempenho.

3.3.1

Resultados para a função Ackley 2 (F1)

O menor erro de cada uma das combinações de operadores quânticos em termos da melhor solução encontrada, durante as 200 gerações pode ser visto na figura 3-10.

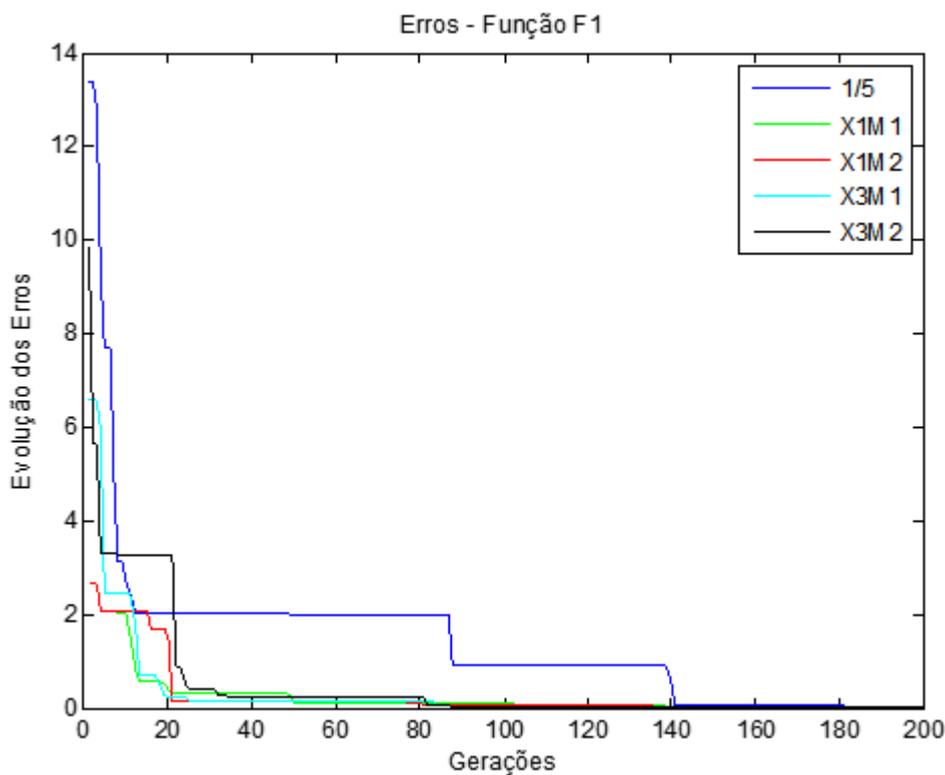


Figura 3-10: Erros durante a otimização da função F1.

Vale destacar que o operador de *crossover two-point* quântico não foi utilizado para esta função, visto que cada cromossomo tem apenas 2 genes, por conta da dimensão da função a ser otimizada.

Analisando-se o gráfico, percebe-se uma convergência antecipada dos algoritmos que utilizaram os operadores de *crossover* e mutação quânticos em relação ao algoritmo que utilizou a regra do 1/5. Entretanto, todos apresentaram resultados satisfatórios, com erros absolutos menores do que 0,04, conforme a tabela 3-4.

Tabela 3-4: Resultados da otimização para a função F1.

Atualização da população	Erro Médio	Desvio Padrão	DPR(%)
Regra do 1/5	0,030929	0,191226	99,64047
X1M1	0,000690	0,049570	83,88763
X1M2	0,001346	0,060725	91,73616
X3M1	0,000111	0,099475	0
X3M2	0,000147	0,124796	24,4129

Os desvios-padrão obtidos foram pequenos, indicando que os modelos apresentaram soluções consistentes e não aleatórias. Cabe destacar o desempenho dos modelos com *crossover* aritmético que tiveram os dois melhores desempenhos.

3.3.2

Resultados para a função Colville (F2)

A partir desta função, todas as combinações entre os operadores foram testadas e a evolução do menor erro para cada uma das configurações testadas pode ser vista na figura 3-11 a seguir.

Analisando o gráfico, nota-se que a maior evolução, em termos das condições iniciais, foi obtida através da combinação X2M2. Já a combinação X1M1 não apresentou uma evolução satisfatória nas gerações iniciais, mas convergiu junto com os demais no final do processo. O desempenho médio das simulações feitas para esta função pode ser visto na tabela 3-5.

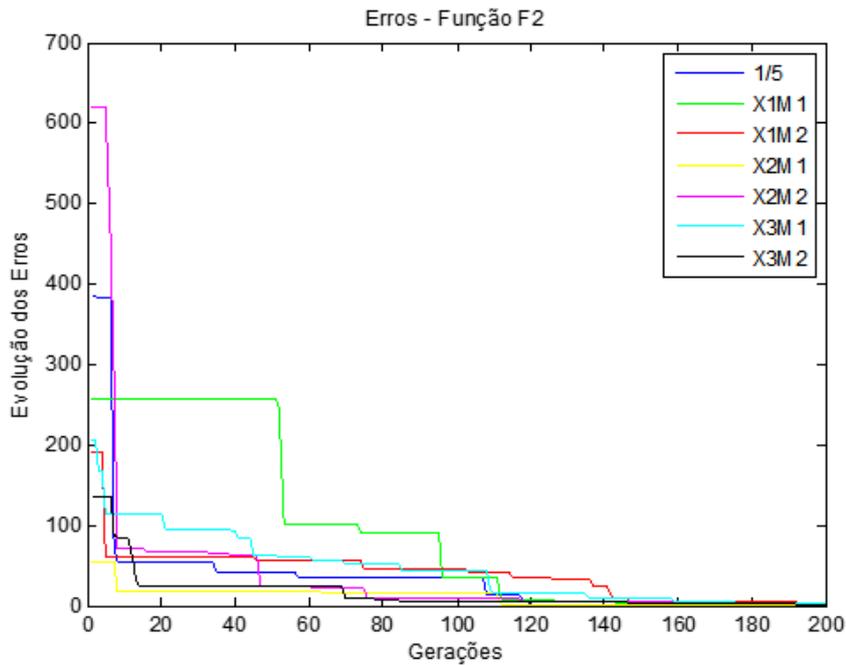


Figura 3-11: Erros durante a otimização da função F2.

Tabela 3-5: Resultados da otimização para a função F2.

Atualização da população	Erro Médio	Desvio Padrão	DPR(%)
Regra do 1/5	1,597845	0,648511	46,8722
X1M1	2,863011	1,038765	70,34941
X1M2	2,167044	0,311371	60,82682
X2M1	1,26094	0,109815	32,6772
X2M2	1,229249	1,026796	30,9416
X3M1	2,446607	0,416223	65,30297
X3M2	0,8489	0,261348	0

Para esta função, a combinação que obteve os melhores resultados foi a X3M2, com um erro inferior a 1 unidade. As combinações X2M2 e X2M1 também apresentaram bons resultados, com erro médio inferior a 1,5 unidades, embora o desvio-padrão da primeira tenha ficado acima de 1.

3.3.3

Resultados para a função Dolan (F3)

Para esta função, as mesmas combinações anteriores foram avaliadas. Os resultados são apresentados a seguir. Nota-se uma convergência mais amena

do que nos testes anteriores, o que pode ser explicado pela complexidade da função e sua maior dimensionalidade, como pode ser visto na figura 3-12.

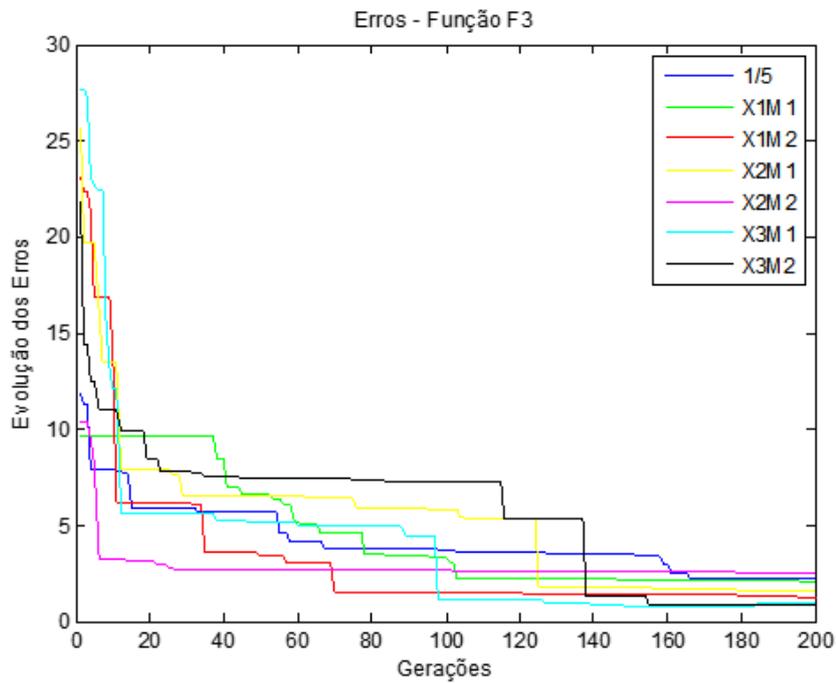


Figura 3-12: Erros durante a otimização da função F3.

O destaque de desempenho fica por conta do *crossover* aritmético quântico que obteve os dois melhores resultados, apesar do desvio padrão acima dos demais operadores, conforme aponta a tabela 3-6.

Tabela 3-6: Resultados da otimização para a função F3.

Atualização da população	Erro Médio	Desvio Padrão	DPR(%)
Regra do 1/5	2,214009	0,172755	62,87597
X1M1	2,057342	0,287036	60,04898
X1M2	1,183285	0,398173	30,53837
X2M1	1,587036	0,379451	48,20979
X2M2	2,461204	0,113847	66,60459
X3M1	0,928983	0,475486	11,5238
X3M2	0,821929	0,355895	0

3.3.4

Resultados para a função Alpine 2 (F4)

No caso da função Alpine 2, alguns algoritmos apresentaram dificuldade de convergência em função da maior dimensionalidade do espaço de entrada (dimensão 7). A configuração X1M1, por exemplo, só conseguiu obter soluções razoáveis ao final do processo evolutivo, conforme pode ser visto na figura 3-13.

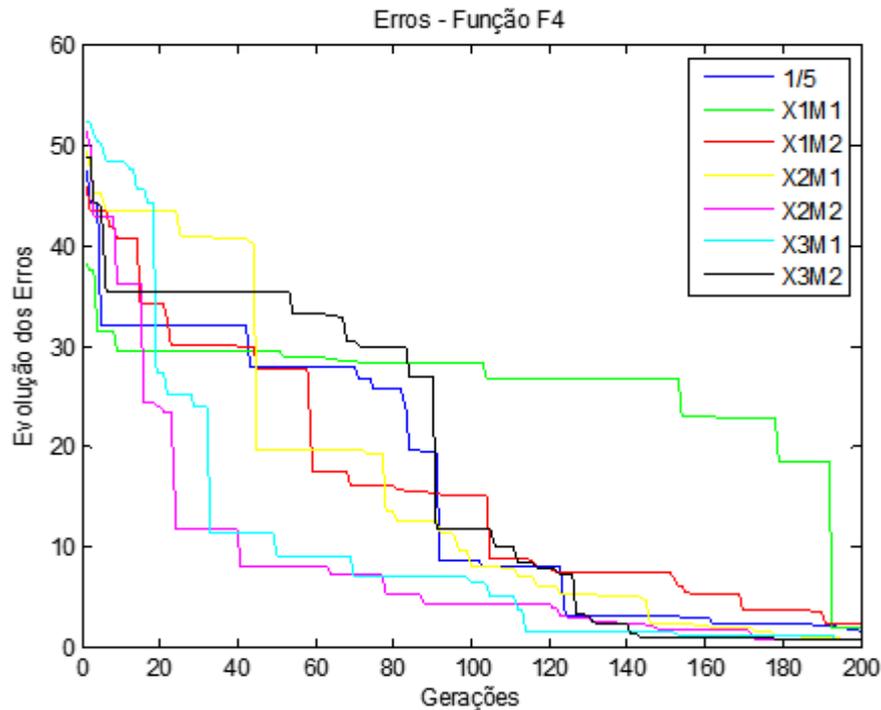


Figura 3-13: Erros durante a otimização da função F4.

Esta configuração (X1M1) foi a que apresentou o pior desempenho, sendo a única com um erro médio superior a 2 pontos.

Tabela 3-7: Resultados da otimização para a função F4.

Atualização da população	Erro Médio	Desvio Padrão	DPR(%)
Regra do 1/5	1,497884	1,314794	64,82543
X1M1	1,783582	0,59882	70,45977
X1M2	2,291276	1,2034	77,00521
X2M1	0,846845	1,554592	37,78392
X2M2	0,57069	1,06626	7,677657
X3M1	0,526874	1,360695	0
X3M2	0,724796	1,551187	27,30728

Para esta função, os desvios-padrão encontrados foram superiores aos das demais funções, o que já era esperado diante do aumento da complexidade da função objetivo. O melhor desempenho foi apresentado pela configuração X3M1.

3.3.5

Comparação dos operadores

Ao longo das pesquisas sobre algoritmos evolutivos, diversos operadores genéticos foram apresentados e testados para as mais diversas aplicações. Esta é uma prova de que a utilidade de um determinado operador depende do problema a ser resolvido, bem como dos parâmetros associados a este. Desta forma, a comparação entre operadores deve ser tratada com cautela e as configurações de parâmetros devem ser testadas em suas diversas combinações possíveis.

Visando comparar entre si as diversas formas propostas de atualização da população quântica, a tabela 3-8 apresenta o desempenho médio da regra do 1/5 e de cada uma das combinações de operadores quânticos testados nas otimizações realizadas.

Tabela 3-8: Desempenho médio das configurações testadas.

Atualização da população	Erro Médio	Desvio Padrão	DPR(%)
Regra do 1/5	1,335167	0,581822	68,55352
X1M1	1,676156	0,493548	71,18645
X1M2	1,410738	0,493417	65,02664
X2M1	1,231607	0,681286	39,55697
X2M2	1,420381	0,735634	35,07461
X3M1	0,975644	0,58797	19,20669
X3M2	0,598943	0,573307	12,93004

Pode-se notar que a melhor combinação, em média, foi a configuração X3M2, com *crossover* aritmético quântico e mutação aritmética quântica, tendo um desvio percentual médio em relação às melhores soluções de apenas 12,93%, enquanto que a regra do 1/5 apresentou uma DPR de aproximadamente 68%.

A combinação X1M2, com *crossover one point* quântico e mutação aritmética quântica foi a combinação que apresentou a menor variação média entre as testadas, indicando robustez das soluções encontradas.

Outro modelo que apresentou bom desempenho foi o que utilizou *crossover* aritmético quântico combinado à mutação *creep* quântica, tendo o melhor desempenho em duas das 4 funções utilizadas como *benchmark*.

Portanto, o modelo NCoQ utilizará os operadores de *crossover* aritmético quântico e mutação aritmética quântica para a realização dos próximos testes, devido aos ganhos de desempenho apresentado por estes.

No próximo capítulo são apresentados os resultados para as aplicações feitas com o modelo NCoQ.

4

Estudos de Casos

Neste capítulo serão apresentados os resultados obtidos pelo modelo NCoQ em três estudos de caso envolvendo multi-agentes cooperativos: dois problemas *benchmarks* (presa-predador e *multi-rover*) e uma aplicação real de otimização de cobertura de telefonia. As seções a seguir detalham a modelagem e os resultados obtidos em cada uma dessas aplicações. Todos os testes foram realizados utilizando-se um computador com processador Intel Core i5 com velocidade de 2.20 GHz e 8 Gb de memória RAM.

Ressalta-se que todos os testes foram em ambientes simulados e seguiram a mesma dinâmica: inicialmente as equipes são treinadas para o ajuste dos parâmetros das redes, de acordo com a função de avaliação do problema a ser tratado. Em seguida, em um ambiente de testes distinto do anterior, testes de desempenho são realizados a fim de verificar a capacidade de adaptação das soluções encontradas.

4.1

Presa-Predador

O *benchmark* escolhido para testar inicialmente o modelo NCoQ foi o problema presa-predador, um clássico exemplo de SMA e comumente encontrado na literatura. Trata-se do jogo de perseguição e captura em um *grid* normalmente discreto e bidimensional, onde cinco agentes, sendo quatro predadores e uma presa, se movem pelo espaço, não podendo ocupar a mesma posição simultaneamente. O objetivo dos caçadores é capturar a presa e esta é considerada capturada quando é cercada pelos predadores, não tendo assim a possibilidade de se mover. Para um melhor entendimento de como a captura é feita, pode-se consultar a figura 4-1. Cabe enfatizar que a captura pode ocorrer de outras formas, utilizando o canto do mapa. Por exemplo, tendo caçadores nas posições (1,2) e (2,1), uma presa na posição (1,1) seria considerada capturada.

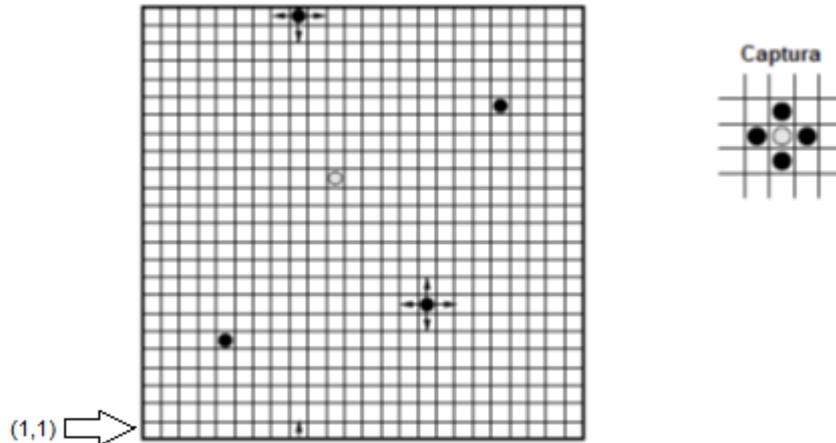


Figura 4-1: *Grid* ortogonal mostrando o esquema do Jogo da Presa/Predador.

O sucesso do problema presa-predador para a avaliação de SMA pode ser explicado por 2 fatores:

- A facilidade de implementação;
- A dificuldade de se encontrar soluções analiticamente, dada a variedade de possibilidades e a necessidade de coordenação dos agentes.

Por exemplo, a tarefa de captura de uma presa não-estacionária é uma tarefa de decisão sequencial. Só é possível avaliar se o time de agentes possui uma boa política de decisão após a execução de uma sequência de passos. Considere que o ambiente de simulação é um mundo toroidal onde presa e predador possuem mesma velocidade. Considere ainda que a presa, durante toda simulação, possua a estratégia de se deslocar na direção oposta ao indivíduo mais próximo.

Ingenuamente, poder-se-ia entender que uma boa alternativa para os caçadores fosse andar sempre na direção da presa de forma independente, ou seja, sem nenhuma coordenação. Neste caso, poderia acontecer como na figura 4-2: a presa nunca seria encurralada por conta da característica toroidal do ambiente. O ideal seria que os agentes aprendessem a “cercá-la” para, em seguida, realizar a captura.

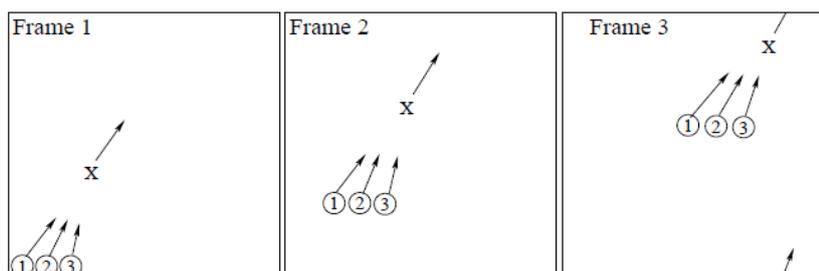


Figura 4-2: Simulação sem coordenação dos predadores.

Como exemplos de trabalhos que estudaram esta forma clássica do jogo e algumas de suas variações pode-se citar (Chainbi, Hanachi & Sibertin-Blanc, 1996), (Corrêa, 2011), (Schrum & Miikkulainen, 2012), (Gao, Givigi & Beaulieu, 2014) e (Mendoza, Vellasco & Figueiredo, 2014).

As variações encontradas estão relacionadas, por exemplo, às características do ambiente (contínuo ou discreto), se o mesmo apresenta obstáculos, as condições de captura, a dinâmica dos movimentos dos agentes, etc. Logo, se faz necessária uma maior descrição do experimento realizado neste trabalho.

4.1.1

Metodologia

As especificações do modelo presa-predador inicialmente implementado para os testes do modelo NCoQ são as seguintes:

- Ambiente composto por um *grid* discreto ortogonal de tamanho 9x9 onde atuam 4 predadores e uma presa;
- Os comportamentos dos agentes predadores evoluem a partir do algoritmo NCoQ;
- Todos os agentes só podem se mover nas direções vertical ou horizontal já que movimentos diagonais não são permitidos. Também é permitido que os agentes fiquem parados. Logo, cada agente possui 5 possibilidades de ação: cima, baixo, direita, esquerda e parado;
- Cada rodada do jogo é composta por um único movimento de cada participante realizados de forma sequencial, sendo que o primeiro movimento é o da presa;
- Não há o conceito de aceleração. Cada movimentação é sempre feita para uma posição imediatamente acima, abaixo, à esquerda ou à direita;
- Cada posição do grid só pode ser ocupada por um único agente;
- Caso algum agente seja orientado para realizar um movimento irregular, como por exemplo, ir para fora do grid ou para alguma posição já ocupada, nesta rodada o agente ficará parado.

A seguir, os comportamentos da presa e do predador, bem como a dinâmica de treinamento, serão detalhados.

4.1.1.1

Comportamento da presa

A inteligência e, conseqüentemente, o comportamento da presa irá depender do estágio de desenvolvimento dos predadores, uma vez que (Gomez & Miikkulainen, 1997) propõem uma abordagem através de um aprendizado incremental, onde os agentes inicialmente são treinados para realizarem tarefas mais simples. Sendo assim, optou-se por uma presa que permanecesse fixa na sua posição inicial durante o processo de treinamento, devendo os caçadores andar em sua direção. Outros comportamentos da presa são analisados durante a fase de testes.

4.1.1.2

Comportamento dos predadores

Para testar a capacidade de aprendizagem por parte dos agentes predadores foi feita a evolução de cada um destes através do algoritmo NCoQ. Neste sentido, o objetivo do algoritmo é encontrar 4 redes neurais recorrentes que co-evoluem paralelamente para obter o máximo de coordenação possível na realização da tarefa de captura da presa. Foram consideradas como variáveis de entrada a posição (x,y) de todos os indivíduos, totalizando $5 \times 2 = 10$ entradas.

Todas as redes neurais possuem a mesma arquitetura recorrente como a descrita no item 3.1.3 e ilustrada na figura 3-4, sendo o número de neurônios inicial um parâmetro fornecido pelo usuário. A única restrição para esse parâmetro é que este seja maior ou igual à dimensão de saída.

Para a saída da rede neural, por conta da natureza discreta do conjunto de ações dos agentes (norte, sul, leste, oeste, parado), foi utilizada a metodologia do neurônio vencedor. Dessa forma, a camada de saída de cada rede possui cinco neurônios, cada um correspondendo a um possível movimento do predador, sendo que a ação a ser realizada corresponde ao neurônio cuja energia de ativação é a maior. A figura 4-3 a seguir ilustra o que foi descrito.

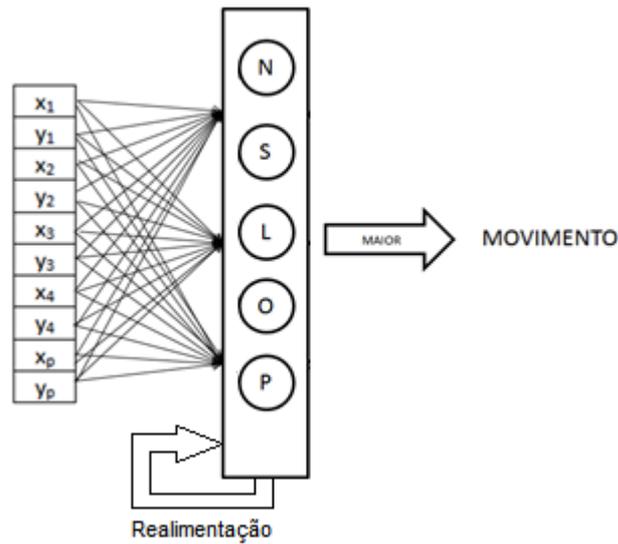


Figura 4-3: Entrada e saída da rede neural utilizada.

A seguir, será detalhada a dinâmica de ajustes dos pesos sinápticos da rede.

4.1.1.3

Dinâmica de treinamento

Para garantir que os caçadores aprendam a caçar a presa e não apenas a chegarem num determinado ponto do *grid*, cada ciclo foi composto por 4 caçadas, com as posições iniciais de cada agente determinadas de maneira aleatória em cada uma delas. Em cada geração, essas posições iniciais são previamente estabelecidas e utilizadas na avaliação de todas as equipes. Em cada caçada, o número máximo de passos permitidos para cada equipe de predadores para a captura é 50. A figura 4-4 a seguir ilustra a dinâmica sequencial da modelagem presa-predador para uma caçada.



Figura 4-4: Dinâmica sequencial de uma caçada.

Uma questão fundamental para o bom funcionamento de um algoritmo genético é a definição da função de avaliação utilizada, pois a chance de um indivíduo sobreviver à geração seguinte está diretamente relacionada com a avaliação recebida. Uma das avaliações mais utilizadas para o problema presa-predador é o número de passos dados pelos agentes até a captura da presa. Entretanto, de acordo com essa avaliação, todos os times que não conseguissem capturar a presa receberiam a mesma nota, no caso o número máximo de passos permitidos. Em um estágio mais avançado da evolução, onde as equipes já conseguem capturar a presa e desejam otimizar essa captura, esta forma de avaliação é válida. Porém, nos experimentos realizados, a utilização do número de passos necessários para a captura da presa não se mostrou adequada, visto que a taxa de captura nas gerações iniciais é muito baixa, havendo assim uma uniformidade da avaliação, o que prejudica o aprendizado. Por exemplo, a figura 4-5 a seguir ilustra dois casos que receberiam a mesma função de avaliação. Admitindo-se a mesma disposição inicial dos agentes no mapa (figura 4-5a), caso as configurações finais de dois cenários sejam as apresentadas nas figuras 4-5b e 4-5c, a avaliação das equipes seria a mesma, uma vez que em ambos os casos a presa não foi capturada ao final dos 50 passos permitidos. Entretanto, o time do cenário 1 obteve um comportamento notoriamente mais adequado do que o time do cenário 2.

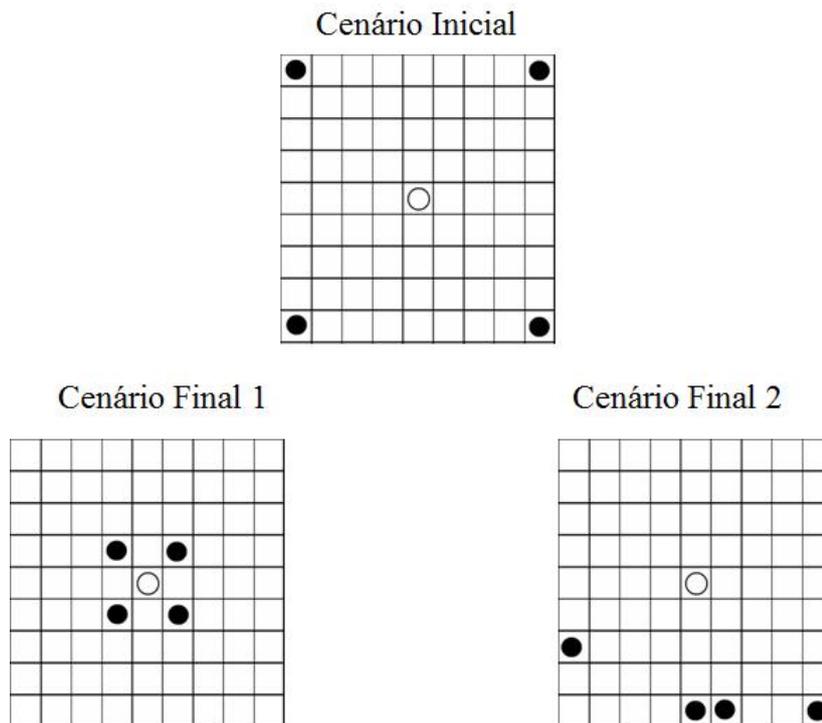


Figura 4-5: Comparação entre cenários.

Neste sentido, optou-se por utilizar uma função de avaliação com duas sentenças: uma para o caso de captura e outra no caso contrário, de forma análoga à proposta por (Yong & Miikkulainen, 2001). A função utilizada pode ser vista na equação a seguir:

$$f(x) = \begin{cases} \frac{d_0 - d_f + 10}{10}, & \text{se a presa não foi capturada} \\ d_0 + (51 - \text{número de passos dados}), & \text{caso contrário} \end{cases} \quad (4.1)$$

onde d_0 é a distância média inicial dos predadores à presa e d_f a distância média final dos predadores à presa. Entende-se aqui como distância a métrica euclidiana entre dois *pixels* do *grid*. Note que $d_0 = 1$ é o menor valor possível para a distância inicial e 11 a maior distância possível num *grid* de dimensões 9x9. Assim, a primeira sentença da função assume valores entre 0 e 2. Já o valor mínimo da segunda função é 2, pois 50 é o máximo de passos que podem ser dados.

Essa função de avaliação foi escolhida por satisfazer os seguintes critérios:

- 1) Caso a presa não seja capturada, a equipe que mais se aproxima da presa é melhor avaliada;

- 2) Caso a presa seja capturada, a equipe que o fizer em menos passos é melhor avaliada. Além disso, a distância inicial também é considerada para não beneficiar situações em que os predadores começam a caçada próximos à presa;
- 3) Uma equipe que consegue capturar a presa é sempre melhor avaliada em relação a uma equipe que não consegue a captura.

No próximo item, os resultados de treinamento do modelo NCoQ são apresentados.

4.1.2

Resultados dos treinamentos com o modelo NCoQ

A escolha dos parâmetros é fundamental para o funcionamento apropriado de um modelo neuro-evolutivo. Por conta disso, diversas configurações de parâmetros foram utilizadas para o treinamento, sendo a que obteve os melhores resultados podendo ser vista na tabela 4-1 a seguir. O tempo de treinamento no melhor desempenho obtido foi de aproximadamente 18 horas.

Tabela 4-1: Parâmetros utilizados no treinamento.

Parâmetros	Faixa de valores testados	Valores do melhor treinamento
Número de subpopulações	4	4
Número de indivíduos	15 – 40	20
Pulsos Quânticos	[-1,1]	[-1,1]
Gerações	60 - 100	100
Número médio de avaliações por indivíduo	10	10
Decaimento permitido <i>lesion</i>	0.05 – 0.2	0.10
Taxa de mutação	0.02 – 0.1	0.02
Taxa de <i>crossover</i>	0.70 – 0.90	0.80
Número de neurônios na camada recorrente	20 – 35	35
Intervalo de atuação do operador <i>lesion</i>	10–20 gerações	10 gerações
Número de colapsos	3 – 5	3

Uma das características desejáveis de um algoritmo é a sua capacidade de convergência. Em um algoritmo evolutivo, é desejável que, durante a evolução, as populações sucessoras, em média, apresentem um melhor desempenho que as

anteriores, o que não aconteceria, por exemplo, numa busca aleatória. Para a análise do modelo NCoQ neste quesito, pode-se observar a figura 4-6 que apresenta a evolução da média dos times clássicos durante a evolução da melhor solução encontrada.

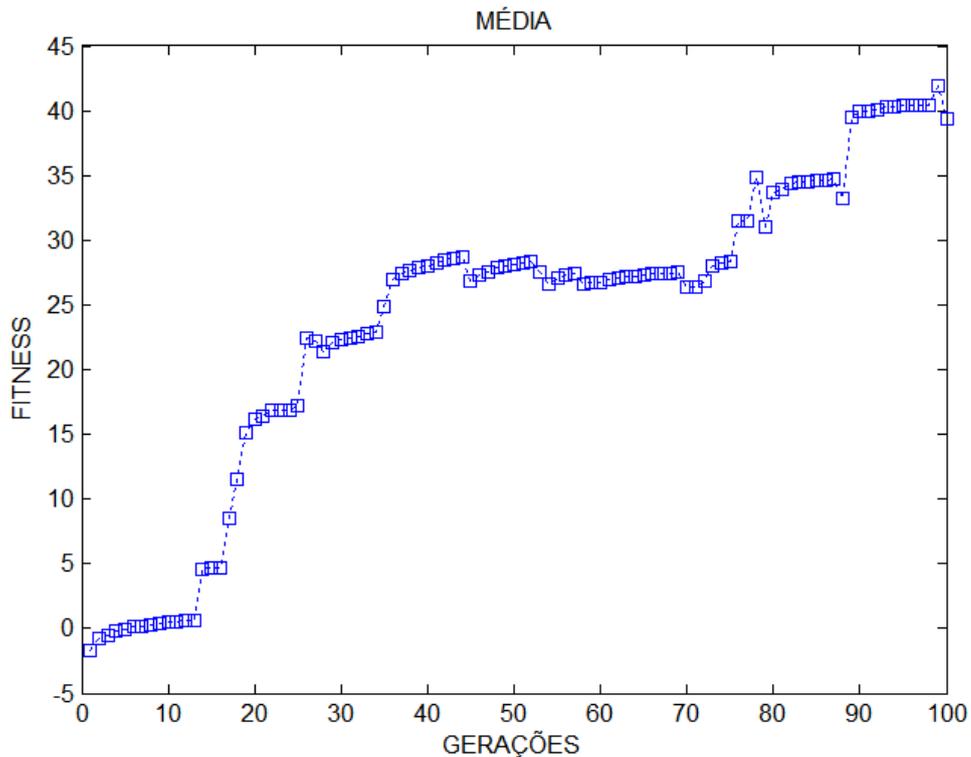


Figura 4-6: Avaliação Média dos Times clássicos.

Note que nas avaliações iniciais o fitness é próximo de zero, o que aponta um insucesso no objetivo de captura da presa dentro do limite de passos permitidos. Inclusive pode-se observar inicialmente uma avaliação negativa, indicando que os predadores estavam se afastando da presa. Este período pode ser considerado como de adaptação ao ambiente. Percebe-se também que entre as gerações 40 e 70 do treinamento a média das avaliações se mantém estagnada, retomando o crescimento somente após a 70ª geração.

Uma das medidas mais utilizadas para a avaliação de uma equipe de predadores na caçada de uma presa é o número de passos que a mesma leva para realizar a captura. Visando comparar o desempenho da equipe através desse parâmetro, o número de passos da melhor equipe em cada geração foi armazenado durante o processo evolutivo e pode ser visto no gráfico a seguir.

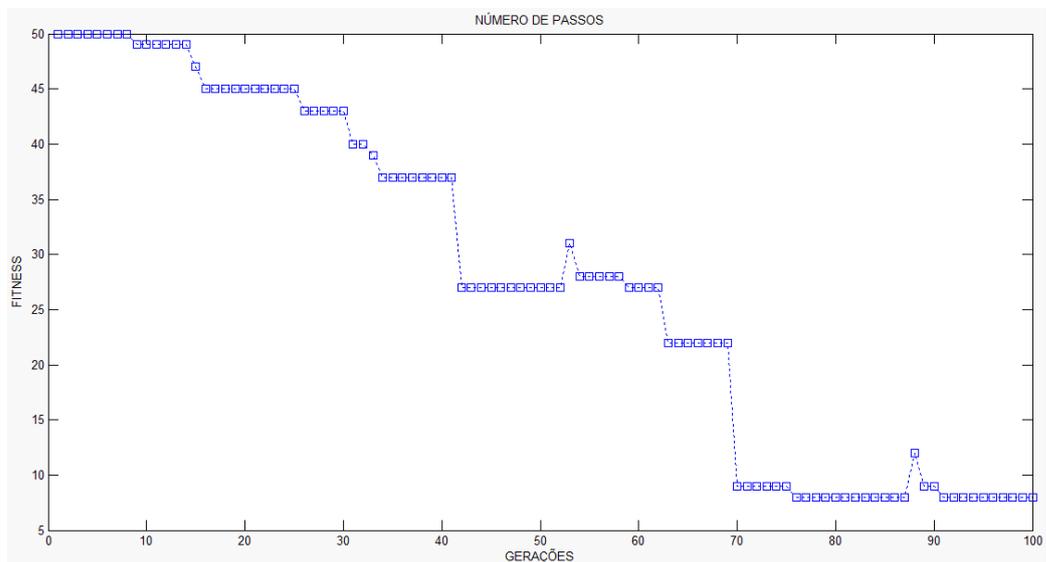
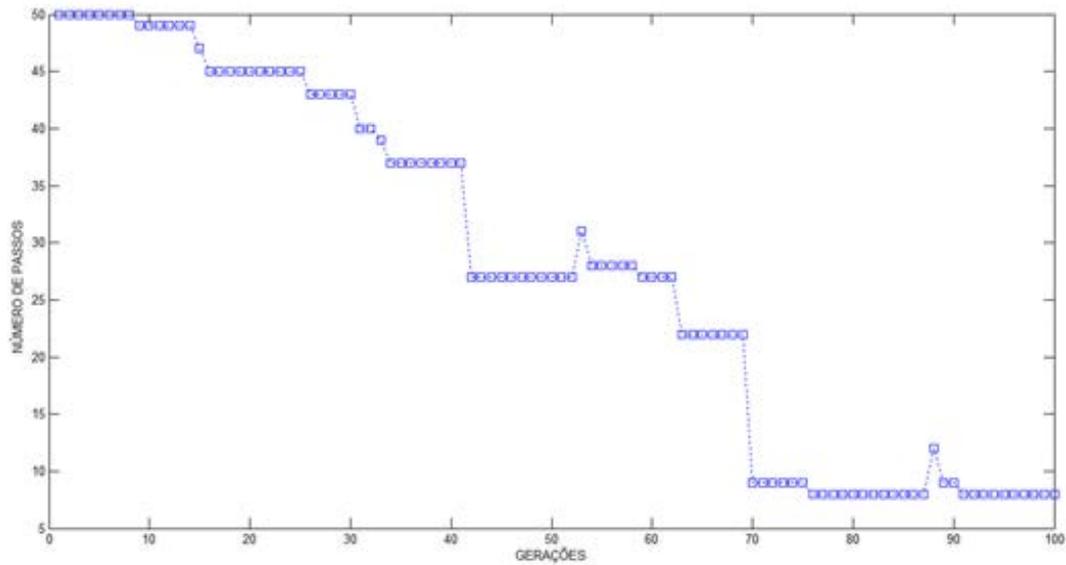


Figura 4-7: Número de passos necessários para a captura durante a evolução.

Percebe-se que nas gerações iniciais a captura não é obtida dentro do limite de 50 passos, sendo a presa capturada pela primeira vez apenas na 9ª geração, com um número de passos ainda muito elevado. Ao longo do processo, a equipe vai se ajustando e a partir da 70ª geração já consegue a captura com um número de passos inferior a 10. Em algumas gerações, há um aumento do número de passos da captura, por conta do operador *lesion* que será analisado mais adiante.

A figura 4-8 exibe a evolução do comportamento do *fitness* ao longo do processo de treinamento e indica que a convergência foi obtida após 70 gerações aproximadamente.

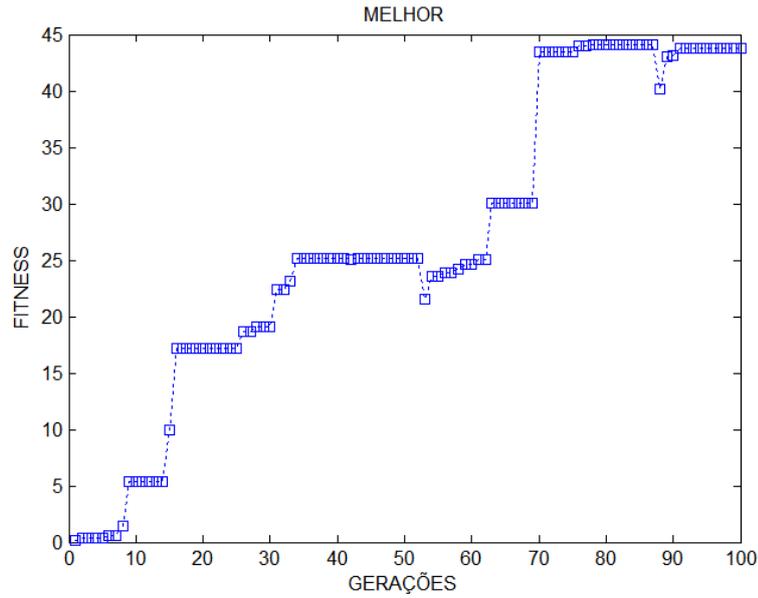


Figura 4-8: Gráfico da evolução do fitness da melhor avaliação.

Há uma pequena queda por volta das 50ª e 90ª geração, decorrente da taxa de queda permitida na avaliação durante o operador *lesion*. Com a retirada dos neurônios, a avaliação tem um pequeno recuo e com os ajustes feitos nas gerações seguintes, consegue recuperar o patamar anterior.

Para efeito de monitoramento deste operador, pode-se observar o gráfico que informa o número de neurônios para cada especialista ao longo das gerações para a melhor solução obtida.

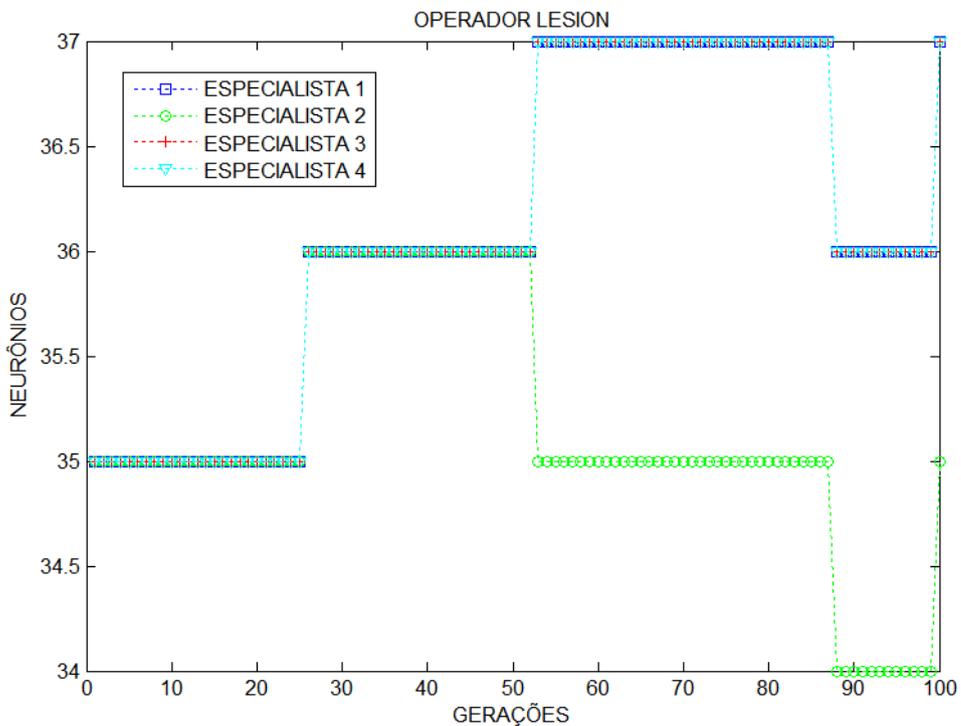


Figura 4-9: Número de neurônios na camada recorrente durante o treinamento.

Pela figura 4-9, nota-se que o operador *lesion* foi acionado 4 vezes ao longo do processo de treinamento, mais especificamente nas gerações 26, 53, 87 e 99.

Na primeira atuação, o operador acrescentou um neurônio em cada agente. Porém, na geração 53, retirou um neurônio do especialista 2 e adicionou um neurônio aos demais. Após nova estabilização da avaliação, o operador retira um neurônio, tendo assim uma pequena queda na avaliação da melhor solução na geração 87, que pode ser vista na figura 4-8. Entretanto, ao final do processo ele volta à condição anterior. Assim, a melhor rede avaliada terminou com 37 neurônios para todos os especialistas, exceto o especialista 2 que terminou a evolução com 34 neurônios.

Sendo o sistema multi-agente implementado homogêneo, por conta das características idênticas dos caçadores, era de se esperar que as redes mantivessem a mesma arquitetura. O que pode explicar o surgimento de uma arquitetura distinta e, conseqüentemente, um comportamento distinto de um dos agentes é a própria estratégia de captura desenvolvida pelos agentes. Este comportamento distinto de um dos indivíduos foi analisado por (Yong & Miikkulainen, 2001) que identificou um agente bloqueador, enquanto que os demais perseguem a presa.

4.1.3

Resultados dos testes com o modelo NCoQ

Com o objetivo de verificar a eficácia do treinamento e a capacidade de coordenação dos agentes, foram realizados testes análogos ao proposto por (Yong & Miikkulainen, 2001).

Nos testes os caçadores começam nos cantos do grid e a presa ocupa as posições numeradas, como na figura 4-10, num total de 9 caçadas. O comportamento da presa também foi simulado para 3 níveis distintos:

- Nível 1: presa fica estática;
- Nível 2: presa anda aleatoriamente;
- Nível 3: presa anda na direção oposta ao predador mais próximo.

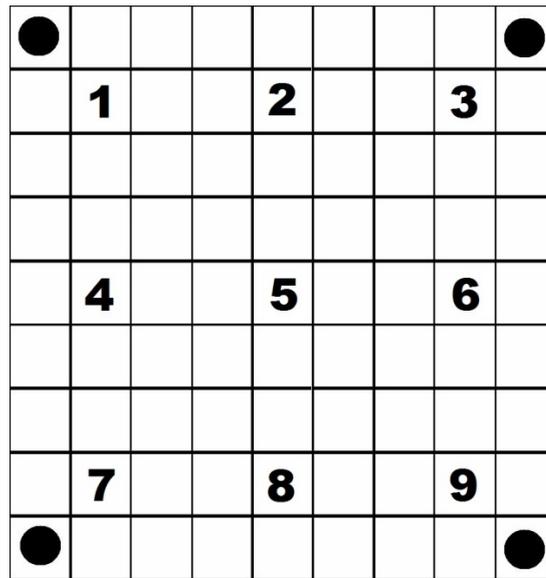


Figura 4-10: Posições iniciais da presa na fase de testes.

Para efeito de comparação, também foram modelados um modelo neuro-evolutivo clássico e um modelo AEIQ-R, cujas metodologias podem ser consultadas em (Yao & Liu, 1997) e (Cruz, 2007), respectivamente.

Os intervalos de parâmetros testados, bem como os valores efetivamente utilizados nestes testes, podem ser vistos nas tabelas 4-2 e 4-3 a seguir.

Tabela 4-2 Configuração dos parâmetros utilizados no modelo de Neuro-Evolução Clássica

Parâmetros	Faixa de valores testados	Valores do melhor treinamento
Número de indivíduos	15 – 40	20
Gerações	60 - 100	100
Taxa de mutação	0.02 – 0.1	0.02
Taxa de <i>crossover</i>	0.70 – 0.90	0.80
Número de neurônios na camada recorrente	20 – 30	30

Tabela 4-3: Configuração dos parâmetros utilizados no modelo AEIQ-R.

Parâmetros	Faixa de valores testados	Valores do melhor treinamento
Número de indivíduos	15 – 40	20
Pulsos Quânticos	[-1,1]	[-1,1]
Gerações	60 - 100	100
Número médio de avaliações por indivíduo	10	10
Decaimento permitido <i>lesion</i>	0.05 – 0.2	0.10
Taxa de <i>crossover</i>	0.70 – 0.90	0.80
Número de neurônios na camada recorrente	20 – 35	35
Intervalo de atuação do operador <i>lesion</i>	10–20 gerações	20 gerações
Número de colapsos	3 – 5	4

Os resultados obtidos nos testes podem ser vistos na tabela 4-4.

Tabela 4-4: Resultados dos testes.

Presas Nível 1 - Parada			
	Capturas Realizadas	Percentual de captura	Número médio de passos
NE Clássica	6	66,67%	24,41
AEIQ-R	8	88,89%	18,56
NCoQ	9	100%	13,33

Presas Nível 2 - Aleatória			
	Capturas Realizadas	Percentual de captura	Número médio de passos
NE Clássica	5	55,56%	36,15
AEIQ-R	7	77,78%	24,14
NCoQ	9	100%	16,29

Presas Nível 3 - Fugindo do mais próximo			
	Capturas Realizadas	Percentual de captura	Número médio de passos
NE Clássica	1	11,11%	48,05
AEIQ-R	5	55,56%	31,05
NCoQ	8	88,89%	21,05

Com a presa estagnada em sua posição inicial (nível da presa 1), o modelo NCoQ obteve melhor desempenho que os demais, superando o modelo AEIQ-R em uma captura. Além disso, o modelo NCoQ precisou de aproximadamente 5

passos a menos, em média, do que o AEIQ-R para a realização da captura. Já o modelo de neuro-evolução clássica conseguiu capturar a presa em apenas 5 das 9 perseguições. Esse fato pode ser explicado pela dificuldade de convergência deste algoritmo para este tipo de problema.

O modelo NCoQ também conseguiu capturar a presa nas 9 perseguições em que esta se deslocou aleatoriamente pelo ambiente (nível 2), porém, em média, 3 passos a mais foram necessários para que isso ocorresse. Isso representa um aumento de aproximadamente 22% em relação ao cenário anterior. Entretanto, o aumento médio no número de passos para o modelo NE clássico e o AEIQ-R, foram de 48% e 30%, respectivamente. Além disso, estes modelos não conseguiram manter o percentual de captura do nível anterior, tendo uma captura a menos para cada um.

Entretanto, é no terceiro cenário, onde a presa apresenta um comportamento mais complexo fugindo do caçador mais próximo, que o modelo NCoQ se destaca dos demais. Apesar de não ter conseguido capturar a presa em todas as perseguições, falhando em uma delas, o modelo NCoQ foi o único que apresentou comportamento satisfatório, precisando em média de 21 passos para a realização da captura. O desempenho do modelo NE clássica para este cenário foi insuficiente, tendo conseguido apenas uma captura. Já o modelo AEIQ-R teve um desempenho razoável, capturando a presa em 5 das 9 caçadas, precisando para isso de 31 passos, 47% a mais do que o número de passos necessários do modelo NCoQ.

Cabe ressaltar que os modelos utilizados na comparação acima são neuro-evolutivos. Os resultados apresentados na 5 visam comparar o modelo NCoQ com outras metodologias encontradas na literatura, como, por exemplo, *Reinforcement Learning*. Os trabalhos de (Corrêa, 2011) e (Mendoza, Vellasco & Figueiredo, 2014) foram escolhidos por utilizarem versões semelhantes do jogo presa-predador para o caso da presa nível 1, ou seja, estática. Estes estudos apresentam modelos de aprendizagem por reforço multi-agente neuro-fuzzy hierárquicos com distintos mecanismos de coordenação.

Foram feitas 1000 simulações, onde presa e predadores largaram de posições aleatórias. O número médio de passos para cada metodologia pode ser visto a seguir.

Tabela 4-5: Desempenho de modelos RL em relação ao modelo NCoQ.

Metodologia	Número médio de passos para a captura	Variação Percentual
RL-NFHP-MA coordenação implícita	9,47	+ 37,64%
RL-NFHP-MA central	7,69	+ 11,77%
RL-SMA-NFHP-CG	7,21	+ 4,79%
NCoQ	6,88	-
RL-SMA-NFHP- <i>Market driven</i>	5,94	- 13,6%

Nota-se que o modelo NCoQ apresenta desempenho inferior apenas ao modelo RL-SMA-NFHP-*Market driven*. Porém, os outros três modelos necessitam de, em média, 18% a mais de tempo para a captura. Isso comprova a robustez do modelo, mesmo quando comparado a outras metodologias de sucesso na literatura.

Vale ressaltar que o tempo de treinamento do modelo RL-SMA-NFHP-*Market driven*, utilizando um computador equivalente foi de 27 horas, 50% a mais do que o necessário para o modelo NCoQ. Já para o modelo RL-NFHP-MA-CG, o tempo necessário para efetuar o treinamento foi de 98 horas. Estes modelos precisam de um tempo maior de treinamento, pois são hierárquicos: primeiro os agentes aprendem o seu papel dentro da equipe; depois desempenham ação, sendo indicado para ambientes mais complexos. Por ter desempenho equivalente com menos informações necessárias, o modelo NCoQ pode ser uma boa alternativa.

4.1.4

Análise do modelo para outras variações no ambiente

Visando analisar a consistência das soluções encontradas, mais duas simulações de testes foram feitas utilizando-se o melhor time encontrado durante o treinamento, com as seguintes modificações no ambiente:

- i) aumento do grid discreto, passando do tamanho 9x9 para o tamanho 21x21;
- ii) transformação do ambiente 9x9 discreto em um ambiente 9x9 contínuo, cuja dinâmica de deslocamento será descrita com maiores detalhes adiante.

4.1.4.1

Testes para ambiente discreto 21x21

Um *grid* de dimensões reduzidas e posições iniciais aleatórias pode fazer com que algumas capturas realizadas pela equipe de caçadores ocorram de forma acidental. Sendo assim, para minimizar esta possibilidade e testar a capacidade de generalização da solução encontrada, foi realizada a mesma dinâmica de testes para um ambiente discreto de dimensões 21x21. Os resultados obtidos podem ser vistos na tabela 4-6. Para esta simulação, o número máximo de passos permitidos foi ampliado de 50 para 70.

Tabela 4-6 Resultado dos testes para *grid* 21x21

Nível da presa	NCoQ	
	Percentual de captura	Número médio de passos
1	100%	30,22
2	88,89%	38,00
3	77,78%	44,67

Analisando os resultados obtidos percebe-se uma queda no desempenho dos caçadores. Cabe observar o aumento exponencial no ambiente da tarefa. No cenário 9x9, há um total de 81 posições possíveis, enquanto que no cenário 21x21 existem 441 possibilidades, um aumento de 444,44%. Entretanto, o número médio de passos para a presa nível 1, foi da ordem de 126% superior ao caso 9x9. Outro fator que deve ser considerado é o fato do treinamento ter sido realizado num *grid* de dimensões diferentes do ambiente de testes. Pode-se considerar, portanto que a solução encontrada foi consistente em relação ao aumento da dimensionalidade da tarefa.

4.1.4.2

Testes para ambiente 9x9 contínuo

Diversas aplicações do mundo real acontecem em ambientes contínuos. Neste sentido, visando aproximar o ambiente simulado da realidade, optou-se pela realização de testes em um ambiente contínuo. Porém, algumas adaptações foram necessárias para a realização dos mesmos.

Neste ambiente a presa é considerada capturada quando os 4 caçadores estão a uma distância menor do que uma unidade da mesma. O deslocamento da presa, em cada nível, seguirá a mesma dinâmica anterior com o tamanho dos passos fixados em 1.

Já o movimento do caçador teve que sofrer uma maior adaptação. Ao invés de considerar a saída única do neurônio vencedor, como na figura 4-3, foi feita uma composição destas saídas com o objetivo de obter o deslocamento horizontal (dx) e vertical (dy) de cada caçador, conforme pode ser vista nas equações a seguir:

$$dx = L - O \quad (4.2)$$

$$dy = N - S$$

onde as constantes L, O, N e S são as saídas dos neurônios conforme a figura 4-3. O caçador irá se deslocar no ambiente na direção e sentido do vetor (dx, dy) , porém, com módulo igual a 1. A ação parada neste modelo não é considerada, mas um comportamento próximo pode ser obtido quando a saída da rede for próxima de 0. Os resultados deste modelo podem ser vistos na tabela 4-7. O número máximo de passos para estes testes foi igual a 50.

Tabela 4-7: Resultados dos testes para *grid* 9x9 contínuo.

Nível da presa	NCoQ	
	Percentual de captura	Número médio de passos
1	100%	15,44
2	100%	14,55
3	88,89%	26,00

Comparando estes resultados com a tabela 4-4, nota-se a manutenção no percentual de tarefas bem realizadas, com uma pequena melhora para os cenários onde o nível da presa foi 1 e 3, e uma melhora no nível 2. Logo, pode-se afirmar que os desempenhos em ambientes discretos e contínuos foram satisfatórios e muito semelhantes.

4.1.5

Análise do modelo para co-evolução competitiva

Através da co-evolução competitiva tem-se outra possibilidade de evoluir populações de agentes. Neste caso, tanto os predadores quanto a presa são evoluídos no processo de co-evolução. Desta forma, o predador evolui para pegar a presa e, concomitantemente, a presa evolui para escapar do predador. Para testar o modelo NCoQ sob esta abordagem, o mesmo foi aplicado tanto nas subpopulações de caçadores como na população de presas. A configuração de ambiente utilizada para esta aplicação foi discreta 9x9.

A evolução dos predadores ocorreu da mesma forma que no item 4.1.1.2. e as entradas e saídas da rede que controlam a presa serão as mesmas da figura 4-3. Entretanto, presa e predadores possuem objetivos distintos e uma nova função de avaliação da presa se faz necessária, já que esta é melhor avaliada quando se mantém distante dos predadores, obtendo sucesso quando não capturada. Optou-se pelo número de passos dados até a ocorrência da captura como parâmetro de avaliação da presa. Mas, por conta do objetivo da presa em escapar dos caçadores, serão melhor avaliadas as presas que levarem mais tempo para serem capturadas. Logo, o objetivo das presas é maximizar o número de passos durante as caçadas.

Para cada time quântico de caçadores, a ser avaliado usando a função de avaliação dada pela equação 4.1, e os parâmetros ajustados conforme o melhor modelo da tabela 4-1, uma presa é selecionada aleatoriamente de uma população quântica. Após serem colapsados, 4 caçadas são realizadas partindo-se de pontos aleatórios do ambiente, sendo a avaliação média de cada agente repassada ao indivíduo quântico correspondente. Na figura 4-11 pode-se visualizar a dinâmica do repasse da avaliação aos indivíduos quânticos.

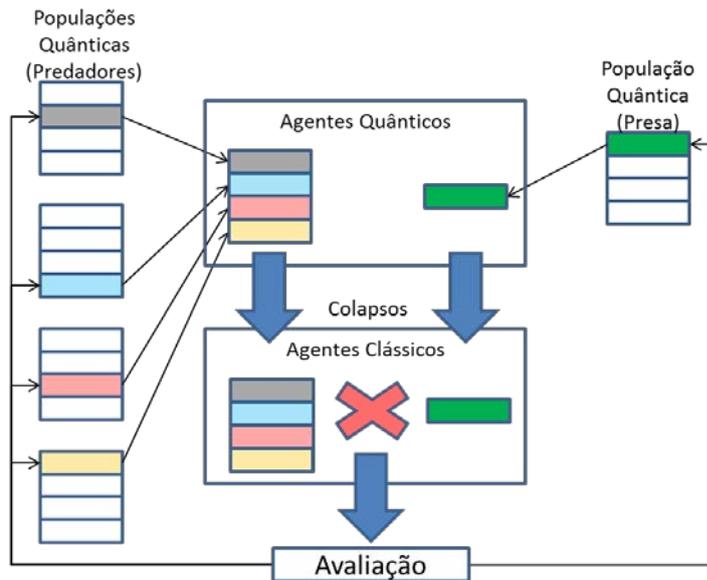


Figura 4-11: Dinâmica de avaliação para co-evolução competitiva.

Ao final de cada geração, os melhores indivíduos clássicos são evoluídos e servirão de parâmetros para atualização da população quântica na geração seguinte. Visando analisar o comportamento da evolução das espécies, um gráfico com o número de passos dos caçadores e da presa durante a evolução pode ser visto na figura 4-12 a seguir.

PUC-Rio - Certificação Digital N° 1022360/CA

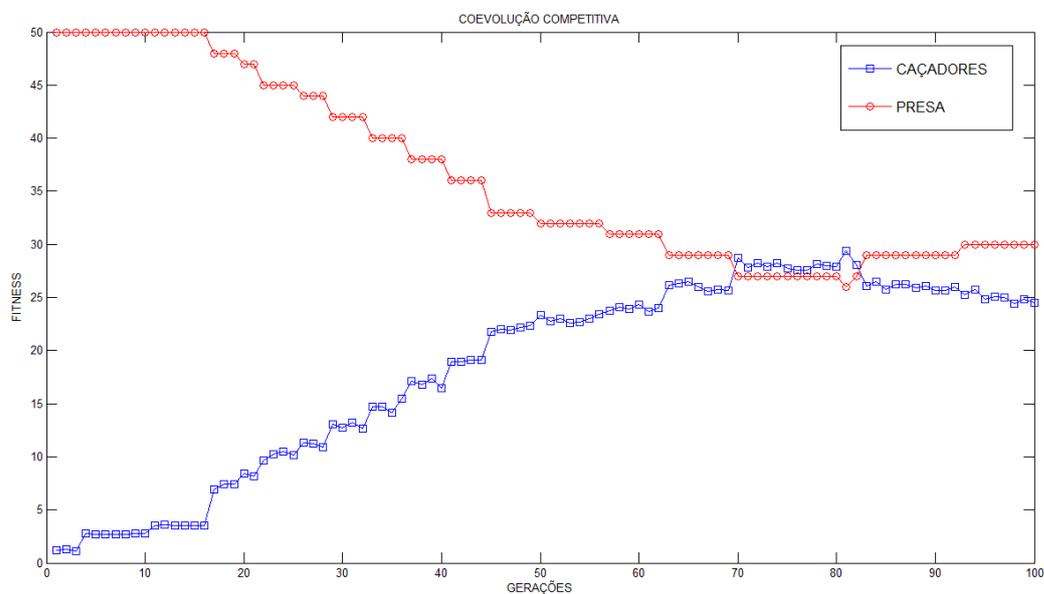


Figura 0-12: Resultado da co-evolução competitiva.

Percebe-se uma evolução menos acentuada por parte dos caçadores em relação aos treinamentos feitos anteriormente, com um ponto de equilíbrio por volta de 30 passos para a captura da presa. Isso pode ser explicado pela

evolução em paralelo da presa, exigindo uma maior coordenação dos caçadores. O operador *lesion* atuou na 10^a, 79^a e 92^a gerações na população das presas, sempre retirando um neurônio, fazendo com que o número de neurônios caísse de 35 para 32 durante o processo evolutivo. Em relação aos caçadores, o operador não foi ativado, por conta da não estabilização da avaliação, mantendo a população com os 35 neurônios originais.

No próximo item, será apresentada mais uma aplicação de coordenação, para a realização de outros testes do modelo NCoQ.

4.2

Multi-Rover

Neste ambiente de simulação, um time de veículos autônomos (sondas ou *rovers*) deve interagir de maneira coordenada para a exploração de um ambiente desconhecido à procura de rubis. Rubis são determinados pontos de interesse espalhados pela superfície de interação que possuem valores distintos a serem coletados. Proposto por (Agogino, 2001), o objetivo desta aplicação multi-agente é simular o comportamento de sondas na exploração de ambientes desconhecidos, como a superfície de outros planetas. Uma ilustração desta tarefa pode ser vista na figura 4-13.

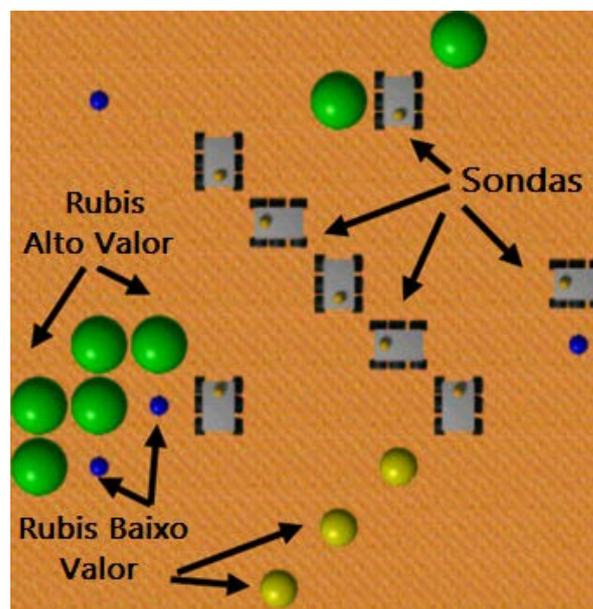


Figura 4-13: Ambiente *Multi-Rover*.

Com o intuito de aumentar a complexidade da coordenação entre os agentes será utilizada, neste trabalho, modelagem semelhante à proposta por (Nitschke, 2009) que difere da original em 3 pontos:

i) Para a detecção de um rubi serão necessárias as participações de pelo menos duas sondas;

ii) As sondas identificarão os rubis através de sensores, assim como na aplicação original. Entretanto, nesta atividade, haverá 3 tipos de sensores que terão custos de energia e aplicações distintos;

iii) Além de considerar o total de pontos coletados pelos agentes, esta tarefa também levará em consideração para a avaliação da equipe o uso racional da energia.

Cabe ressaltar que a opção pela aplicação *multi-rover* também se dá pelo ambiente contínuo de simulação, visando a analisar o desempenho do NCoQ em outro ambiente contínuo distinto ao da presa-predador. O uso de sensores, também simulados, aproxima ainda mais a aplicação de situações reais.

4.2.1

Ambiente

No início de cada simulação, 20 sondas são posicionadas aleatoriamente num *grid* contínuo de dimensão 50x50. Além de desbravar este ambiente, as sondas devem identificar 2375 rubis. Cada rubi necessita da utilização de tipos específicos de sensores, a serem detalhados adiantes, para serem coletados. Os 2375 rubis estão distribuídos no mapa conforme a tabela 4-8 a seguir.

Tabela 4-8: Tipos de rubis e os sensores requeridos para detecção.

Tipo de Rubi	Valor do Rubi	Quantidade	Sondas Requeridas
A	1	500	2 sensores baixos
B	1	500	2 sensores iguais
C	2	500	1 sensor médio + 1 baixo
D	3	500	1 sensor médio + 1 alto
E	4	375	2 sensores altos

Os rubis também são espalhados aleatoriamente no ambiente e, como descritos na tabela acima, irão requerer uma combinação específica de utilização

dos sensores para a identificação de cada tipo. Fica claro que para detectar um rubi é necessária a colaboração de pelo menos duas sondas. Em cada ponto do *grid* só será permitida a presença de um agente. A seguir, serão descritas com maior detalhamento as características específicas das sondas.

4.2.2

Sondas (*Rovers*)

As sondas (*rovers*) irão compor a equipe que deverá coletar os rubis dispostos no ambiente inicialmente desconhecido. Cada sonda será controlada através de uma rede neural recorrente que evolui a partir do algoritmo NCoQ. As entradas serão fornecidas através de 16 sensores, sendo 8 sensores para detectar rubis e outros 8 para detectar outras sondas, dispostos conforme a figura 4-14.

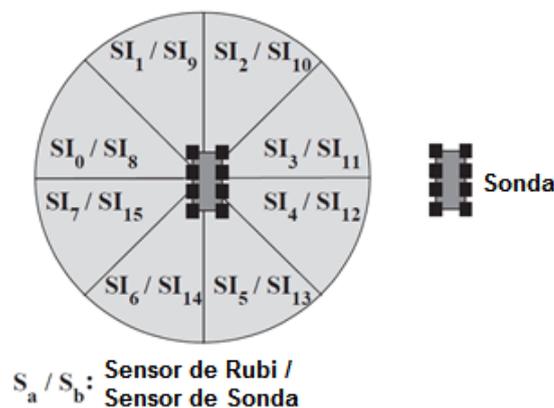


Figura 4-14: Disposição dos sensores nas sondas.

A seguir, o funcionamento dos sensores é explicado em maiores detalhes.

4.2.2.1

Sensores para detecção de rubis

Os sensores para a detecção dos rubis devem ser explicitamente ativados. Esta é uma das ações possíveis da sonda. Caso uma sonda ative o uso dos sensores, todos os oitos sensores são ativados com uma das três características possíveis, conforme a tabela 4-9 a seguir.

Tabela 4-9: Características dos sensores de rubi.

Estado do Sensor	Precisão	Alcance	Custo Energético
Baixo (valor 1)	1,0	4	0,25
Médio (valor 2)	1,0	4	0,5
Alto (valor 3)	1,0	4	1

Precisão indica a probabilidade de um sensor identificar corretamente um rubi. Uma precisão igual a 1 indica que ruídos não serão considerados neste estudo, visando uma maior simplificação. O alcance de todos os sensores deste tipo será igual a 4, ou seja, conseguem enxergar até 4 quadrantes adiante à sonda. Porém, o custo energético varia para cada tipo de sensor e é descontado da energia de cada sonda para cada instante de tempo em que os sensores permanecem ativados. Vale destacar que no instante inicial ($t=0$) o número de pontos de energia é igual a 1000.

A entrada de cada sensor para detecção de rubis será inversamente proporcional à soma das distâncias dos rubis à respectiva sonda, como pode ser visto na equação a seguir:

$$I_{1(q,t)} = \sum_{j \in J_q} \frac{1}{\delta(S_q, R_j)} \quad (4.3)$$

onde,

$I_{1(q,t)}$ = input do sensor 1 (rubis) da q -ésima sonda no tempo t

J_q = quadrantes do alcance do sensor

$\delta(S_q, R_j)$ = distância euclidiana entre a q -ésima sonda e o rubi dentro do alcance.

Cabe ressaltar que quando um rubi é identificado, a pontuação do mesmo é computada e este desaparece do mapa, não podendo ser novamente encontrado por outras sondas em outro instante de tempo.

4.2.2.2

Sensores de identificação de sondas

Os sensores para a detecção de outras sondas possuem, basicamente, duas funções: prevenir colisões entre as sondas e verificar o valor assumido pelo sensor de rubis de uma sonda companheira. Esta segunda função existe por

conta da necessidade de combinação de sensores para a identificação dos rubis, conforme tabela 4-8. A entrada de cada sensor deste tipo será dada pela equação

$$I_{2(q,t)} = \frac{v(S_p)}{\delta(S_q, S_p)} \quad (4.4)$$

Onde:

$I_{2(q,t)}$ = *input* do sensor 2 (sondas) da q -ésima sonda no tempo t

$\delta(S_q, S_p)$ = distância euclidiana entre a q -ésima sonda e a sonda mais próxima desta

$v(S_p)$ = valor assumido pelo sensor de rubis da sonda mais próxima (igual a 1, 2 ou 3, de acordo com a tabela 4-9).

4.2.3

Controle de ação das sondas

Em cada iteração, cada sonda deverá escolher sua própria ação a partir do vetor de entradas composto pelos sinais de cada um dos seus 16 sensores. A saída será fornecida por uma rede neural recorrente e irá determinar, através do neurônio vencedor, uma das 4 ações listadas a seguir:

Ação 0 – Ativar todos os sensores de rubis no parâmetro 0;

Ação 1 – Ativar todos os sensores de rubis no parâmetro 1;

Ação 2 – Ativar todos os sensores de rubis no parâmetro 2;

Ação 3 – Movimentar-se. Neste caso, a direção e o sentido do deslocamento são obtidos a partir das saídas dx e dy de outros dois neurônios específicos.

Na figura 4-15 pode-se ver a arquitetura utilizada. Nesta aplicação, o número de neurônios na camada de realimentação continua sendo um parâmetro a ser evoluído durante o processo.

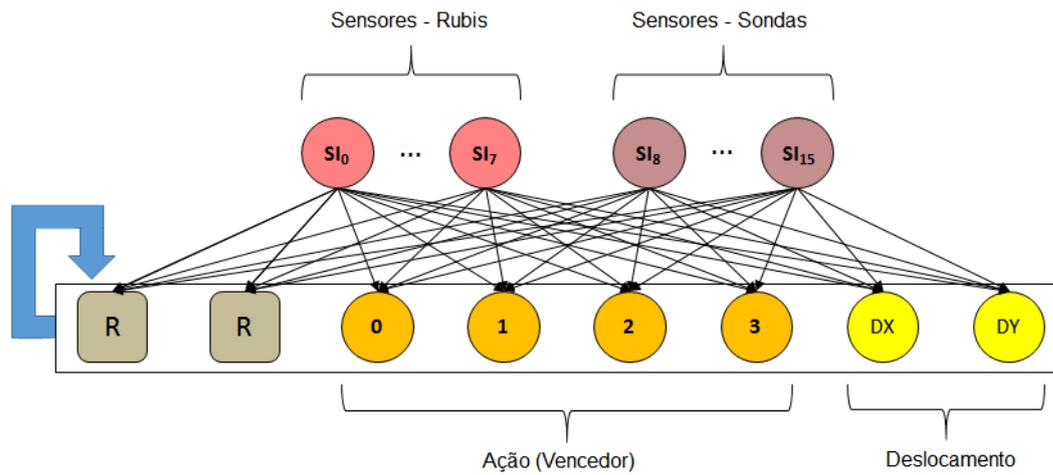


Figura 4-15: Arquitetura da Rede Neural Recorrente para a aplicação *Multi-Rover*.

A evolução dos neuro-controladores será obtida através da aplicação do algoritmo NCoQ e será melhor detalhado no item a seguir.

4.2.4

Dinâmica de treinamento e Neuro-Evolução

Como dito anteriormente, o objetivo do time de sondas é maximizar a pontuação obtida a partir da coleta do maior número de rubis possível dentro do limite de energia pré-estabelecido, considerando o uso racional da mesma. Neste sentido, a função avaliação G para uma simulação a ser maximizada por cada equipe será dada por:

$$G = (\sum_{r \in I} p(r)). \quad (4.5)$$

onde:

I = conjuntos dos rubis identificados;

$p(r)$ = número de pontos referentes ao rubi r ;

A avaliação, portanto, será a pontuação total de rubis recolhidos pela equipe. Na verdade, a avaliação que será repassada aos indivíduos será o desempenho médio da equipe em 5 interações, com posições iniciais de rubis e sondas aleatórias.

Os parâmetros do experimento *Multi-Rover* podem ser vistos na tabela 4-10 a seguir:

Tabela 4-10: Parâmetros da simulação *Multi-Rover* (ambiente).

Parâmetros – Ambiente <i>Multi-Rover</i>	Valores
Alcance do Sensor de Rubis	4
Precisão do Sensor de Rubis	1
Custo do Sensor de Rubis	Variável (Tabela 4-9)
Alcance do Sensor de Sonda	4
Precisão do Sensor de Sonda	1
Energia Inicial da Equipe	1000
Posição Inicial das Sondas	Aleatória
Posição Inicial dos Rubis	Aleatória
Tamanho do ambiente	50x50
Natureza do ambiente	Contínua
Valor Individual dos Rubis	Variável (Tabela 4-8)
Total de Rubis no ambiente	2375
Total de pontos	5000 pontos

Cabe ressaltar que a escolha por esses parâmetros foi orientada pelo estudo de (Nitschke, 2009), visando a uma melhor comparação de resultados, a serem apresentados adiante, assim como o número de 20 sondas utilizado.

Para o treinamento das redes e análise do modelo para esta aplicação, diversas simulações com ajustes distintos de parâmetros foram feitas e duraram, para cada treinamento realizado, 20 horas em média. A faixa de valores testados e a configuração que obteve o melhor desempenho apresentada na tabela 4-11 a seguir.

Tabela 4-11: Parâmetros da simulação *Multi-Rover* (NCoQ).

Parâmetros – NCoQ	Faixa de valores testados	Valores
Número de subpopulações	20	20
Número de indivíduos	10 – 20	10
Gerações	150 – 250	250
Interações	5	5
Pulsos Quânticos	[-1,1]	[-1,1]
Número inicial de neurônios na rede	10 – 20	15
Decaimento permitido <i>lesion</i>		0.10
Taxa de mutação	0.02 – 0.05	0.02
Taxa de <i>crossover</i>	0.60 – 0.80	0.80
Intervalo de atuação do operador <i>lesion</i>	10 – 20 gerações	10 gerações
Número de colapsos (indivíduos clássicos) por indivíduo quântico	3 - 4	3

4.2.5

Resultados – *Multi-Rover*

A evolução do fitness médio das equipes de sondas da população clássica selecionada em cada geração durante a fase de treinamento pode ser observada na figura 4-16 a seguir.

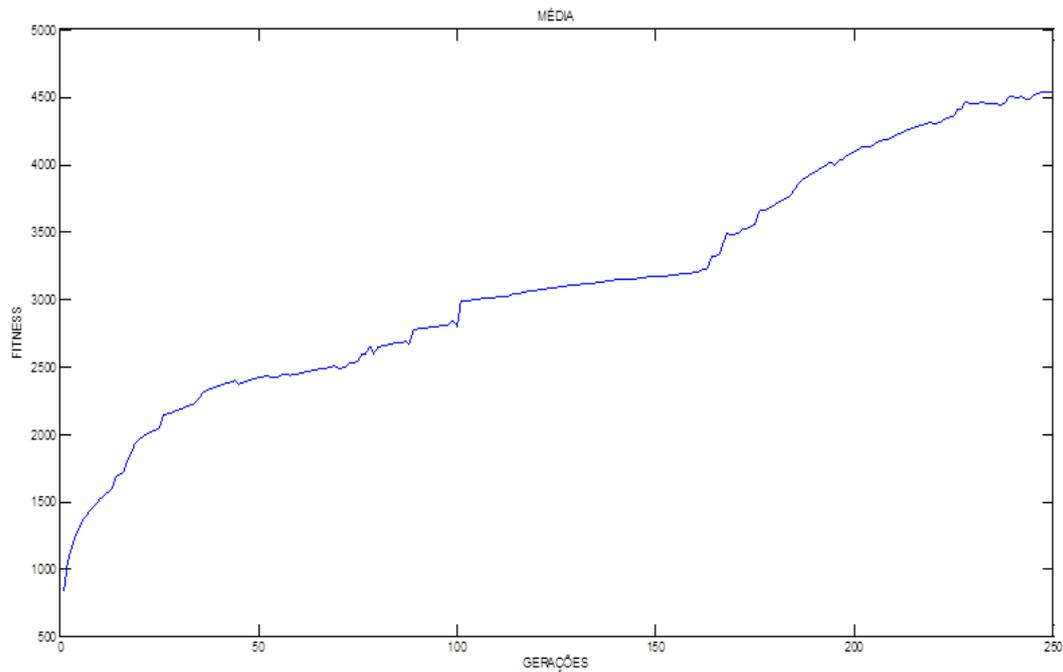


Figura 4-166: Avaliação Média dos Indivíduos Clássicos (*Multi-Rover*).

Pelo gráfico, pode-se notar a evolução média dos indivíduos clássicos, o que demonstra o efetivo funcionamento do modelo em termos de encontrar soluções melhores durante o processo de evolução. Nota-se também uma pequena estabilidade entre as gerações 100 e 150, com a retomada da evolução em seguida.

O gráfico a seguir exibe o *fitness* da melhor equipe durante o processo de evolução.

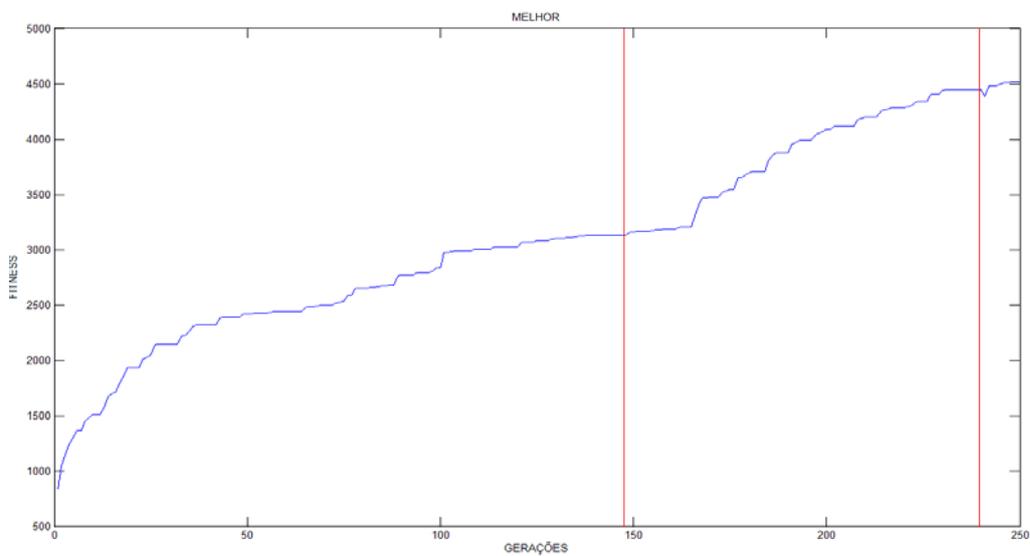


Figura 4.17: Gráfico da evolução do *fitness* da melhor avaliação (*Multi-Rover*).

Pode-se verificar que a avaliação final ficou por volta de 4500, de um total de 5000 pontos a serem coletados no ambiente. As linhas vermelhas marcam as atuações do operador *lesion* que ocorreram nas gerações 148 e 240, onde as avaliações completaram 10 gerações sem melhoria. Há uma pequena queda no desempenho após a segunda atuação do operador, decorrente das retiradas de alguns neurônios dos indivíduos. Entretanto, nas gerações seguintes essa perda é recompensada e a avaliação, inclusive, fica um pouco acima do patamar anterior. Entretanto, pode-se visualizar uma estabilidade no gráfico nas gerações finais, indicando a convergência da evolução.

Para efeito de comparação, os resultados obtidos nesta simulação foram comparados aos obtidos por (Nitschke, 2008). Neste estudo, há a descrição e os resultados para a simulação *Multi-Rover* para as seguintes metodologias:

- HomCNE (Neuro-evolução convencional homogênea): todos os agentes são decodificados por um único genótipo;
- HetCNE (Neuro-evolução convencional heterogênea): cada agente é gerado por um genótipo distinto;
- MESP (Neuro-Evolução ESP Multi-Agente): Extensão do algoritmo ESP para aplicações multi-agente, onde cada rede é formada a partir da evolução de neurônios e suas conexões;
- CCGA (Algoritmo Genético com Co-evolução Cooperativa): utiliza algoritmos genéticos para co-evoluir soluções parciais;
- CONE (Neuro-evolução Coletiva): Evolui subpopulações de neurônios, mas permite recombinação entre as mesmas.

A figura 4-18 exibe a comparação de resultados entre o modelo NCoQ e estas metodologias.

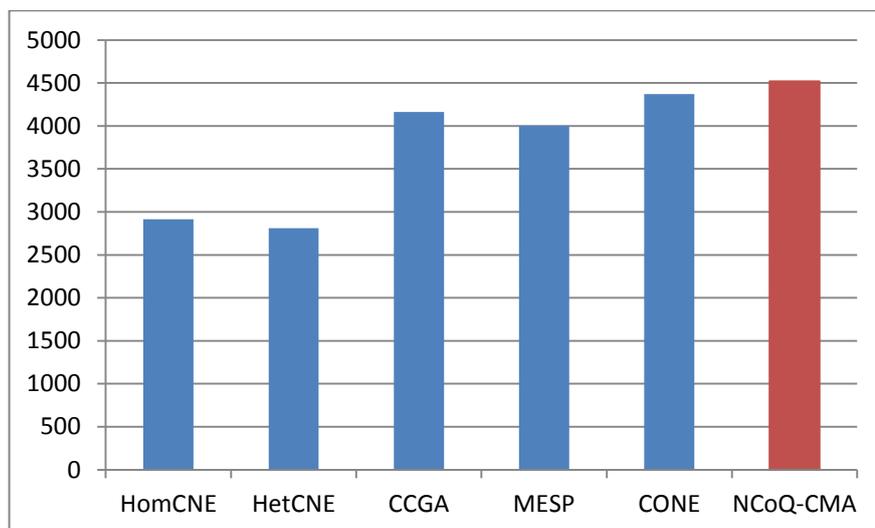


Figura 4.18: Comparação entre metodologias (*Multi-Rover*).

O modelo NCoQ apresentou um desempenho, em média, 28% melhor quando comparado às demais metodologias. Em relação ao modelo de neuro-evolução convencional heterogêneo, por exemplo, o ganho de desempenho chegou a, aproximadamente, 61%. Em relação à neuro-evolução coletiva, o ganho foi de apenas 3%. Entretanto, deve-se destacar que, para este nível de pontuação, não há como ter ganhos relativos consideráveis, pois com 4520 pontos, o modelo NCoQ conseguiu coletar 90,4% dos pontos, sendo a única metodologia a atingir os 90% de eficiência. A tabela 4-12 a seguir exibe os valores e a comparação entre as metodologias.

Tabela 4.12: Comparação entre os modelos no experimento *Multi-Rover*.

Metodologia	Pontos obtidos	(%) do total de pontos	Diferença Percentual para o NCoQ-MA
HomCNE	2916	58,32	55,01
HetCNE	2809	56,18	60,91
MESP	4003	80,06	12,92
CCGA	4161	83,22	8,63
CONE	4371	87,42	3,41
NCoQ	4520	90,40	0,00

Por estas comparações, pode-se perceber a eficiência do modelo NCoQ em tarefas de coordenação, como no caso da aplicação *Multi-Rover*. A próxima seção apresenta a aplicação do modelo NCoQ em um problema real.

4.3

Otimização de Redes de Telefonia

O avanço tecnológico têm permitido um grande acesso à transmissão de dados. É comum o uso de transmissões “ao vivo”, serviços por *streaming* e de outros aplicativos que dependem de uma alta disponibilidade e desempenho da rede de telefonia. Com isso, as operadoras vêm investindo cada vez mais na tecnologia 4G, tendo atingido 41,8% da população nacional coberta ao final de 2014, conforme a tabela 4-13.

Tabela 4-12 Percentual da população coberta pela tecnologia 4G no Brasil.

Fonte: (ANATEL, 2015).

Operadora	Trimestre/Ano								
	4/12	1/13	2/13	3/13	4/13	1/14	2/14	3/14	4/14
Vivo	-	-	18,4	28,3	30	30,3	35,4	35,6	40,6
Claro	0,9	2,5	17	20	25,3	34,1	36,4	36,8	36,8
Oi	-	-	9,4	9,4	23,1	23,1	30,6	30,5	30,5
Tim	-	-	9,4	15,2	23,1	23,1	30,3	30,2	30,2
Nextel	-	-	-	-	-	-	3,2	3,2	3,2
Total	0,9	2,5	18,4	28,7	31,9	36,2	38,5	38,9	41,8

Entretanto, por conta do fator mobilidade, é fundamental mapear as condições do ambiente no qual o sistema será implantado, não sendo uma tarefa fácil a criação de um sistema inteligente que mapeie as condições do ambiente e equalize uma série de parametrizações de maneira ótima, já que os processos de propagação afetam diretamente o desempenho da rede e, conseqüentemente, o tráfego de dados.

Na impossibilidade de medição do nível do sinal, tem-se como alternativa simular o comportamento de propagação das ondas eletromagnéticas através de modelos matemáticos determinísticos ou empíricos.

O objetivo desta simulação é validar a atuação do modelo NCoQ na automação de uma rede móvel, avaliando sua capacidade de ajustar de maneira coordenada os parâmetros das torres de transmissão, visando maximizar a cobertura da rede mantendo uma qualidade mínima de sinal.

4.3.1

Modelos de propagação Okumura-Hata

Podem ser encontrados na literatura uma infinidade de modelos de propagação que simulam ambientes *outdoor*, nos quais a estação móvel encontra-se externa a construções (Valkealahti, 2002) (Parsons *et al.*, 2000) e ambientes *indoor* (Aragon-Zavala, 2008).

Na presente modelagem, optou-se pelo modelo Hata (Hata, 1980), derivado do modelo Okumura (Okumura, 1968), por conta da sua praticidade, já que o mesmo utiliza uma morfologia sem detalhamento. Este modelo utiliza “manchas” morfológicas para diferentes ambientes de propagação, como área suburbana, urbana, urbana densa, rural, etc.

As equações utilizadas para a modelagem podem ser vistas a seguir.

Equação Okumura Hata

(4.6)

$$L_u = 69,55 + 26,16 \log_{10} f - 13,82 \log_{10} h_B - C_H + [44,9 - 6,55 \log_{10} h_B] \log_{10} d$$

Onde,

L_u - valor médio da perda devido ao caminho de propagação para área urbana [dB];

f - frequência [MHz];

h_b - altura efetiva da antena transmissora [m];

d - distância entre a estação de rádio e a base receptora [km];

C_h - fator de correção da antena.

O fator de correção da antena é calculado com base no ambiente de propagação. Para cidades urbanas e urbanas densas, as equações podem ser vistas a seguir.

Fator de Correção para cidade urbana

$$C_{HM} = 0,8 + (1,1 \log_{10} f - 0,7)h_M - 1,56 \log_{10} f \quad (4.7)$$

Fator de Correção para cidade urbana densa

$$C_{HG} = \begin{cases} 8,29(\log_{10}(1,54h_M))^2 - 1,1 & , \text{ se } 150 \leq f \leq 200 \\ 3,2(\log_{10}(11,75h_M))^2 - 4,97 & , \text{ se } 200 < f \leq 1500 \end{cases} \quad (4.8)$$

Onde,

h_M - altura efetiva da antena receptora [m];

f - frequência [MHz].

A dinâmica de otimização será detalhada no próximo item.

4.3.2

Dinâmica de otimização

Um dos fatores que determinam a qualidade e o alcance da rede é a potência disponibilizada por cada antena para as conexões com as redes móveis. Uma maior potência significa maior cobertura, mas implica em uma menor potência excedente responsável pela qualidade do sinal.

Dessa forma, a potência do canal piloto de cada torre deve ser otimizada de acordo com a posição dos celulares e, após os ajustes de cada antena, o percentual de cobertura será avaliado. Portanto, a avaliação do ajuste da potência das antenas p_i será dada pela função:

$$A(p_i) = (\% \text{ de cobertura}). \quad (4.9)$$

O controle da potência piloto de cada torre será feito através de uma rede neural com entradas e arquiteturas de acordo com a figura 4-19, que irá evoluir através do modelo NCoQ.

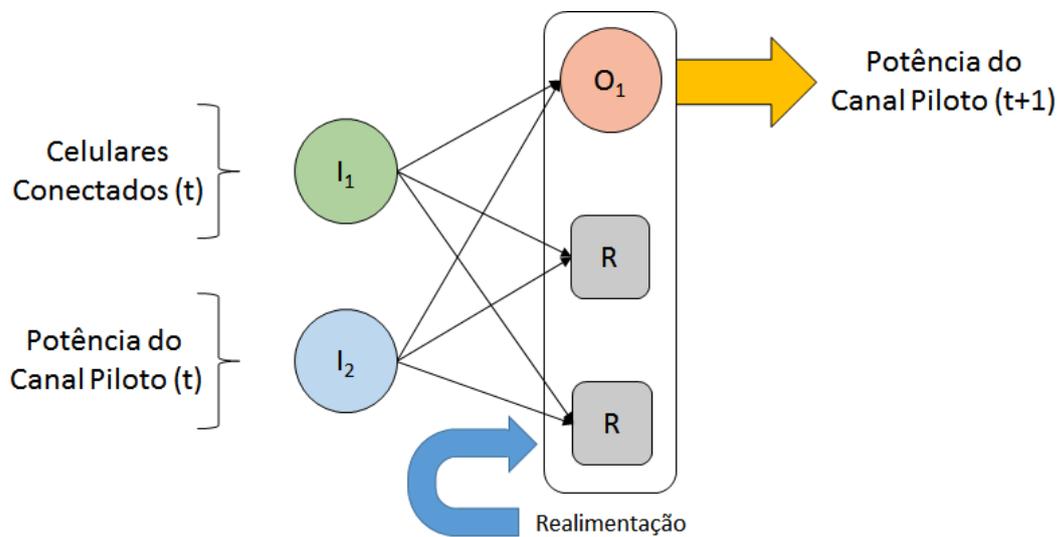


Figura 4.19: Entradas e Arquitetura utilizada para as saídas da Potência Piloto de cada torre.

Os parâmetros do modelo NCoQ que foram utilizados para a realização da otimização podem ser vistos na tabela 4-14 a seguir.

Tabela 4.14: Parâmetros da simulação do modelo NCoQ para cobertura móvel.

Parâmetros – NCoQ	Faixa de valores testados	Valores
Número de subpopulações	50	50
Número de indivíduos	10 – 20	10
Gerações	100 – 200	100
Pulsos Quânticos	[-1, 1]	[-1,1]
Número de entradas (<i>inputs</i>)	2	2
Número inicial de neurônios na rede	4 – 10	4
Decaimento permitido <i>lesion</i>	0.05 – 0.20	0.20
Taxa de mutação	0.05 – 0.10	0.1
Taxa de <i>crossover</i>	0.6 – 0.8	0.80
Taxa de elitismo	0.20 – 0.25	0.25
Intervalo de atuação do operador <i>lesion</i>	10 – 20 gerações	10 gerações
Número de colapsos	5 – 10	10

Vale ressaltar que o número de subpopulações é o número de torres (agentes) que devem ajustar as suas potências respeitando um limite máximo de potência.

Aqui cabe o registro de que outra função de avaliação foi testada, dada pela diferença entre o percentual de cobertura e o percentual médio das potências utilizadas, visando não só otimizar a cobertura diante de uma restrição, mas maximizar também o percentual de potência disponível. Entretanto, os resultados obtidos foram inferiores, com um percentual de cobertura estimado em 78%.

4.3.3

Ambiente de simulação – cobertura móvel

Para a realização do treinamento das redes, uma cidade de dimensões retangulares, 50 km por 30km, e de urbanização densa foi simulada. Nela foram disponibilizadas aleatoriamente um conjunto de 2000 celulares e 50 antenas. O mapa desta cidade simulada, com suas torres (vermelho) e móveis (azul), pode ser visto na figura 4-20.

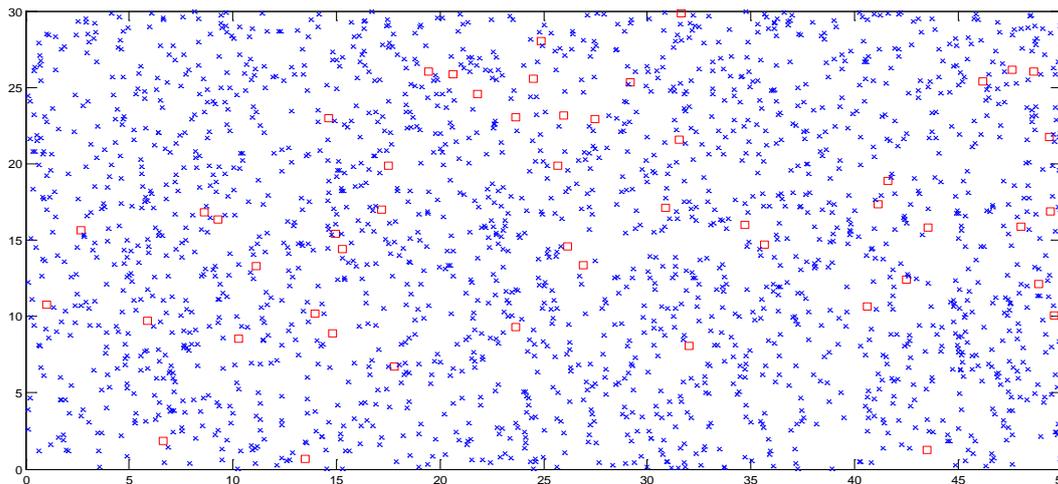


Figura 4.20: Mapa do ambiente simulado.

Outros parâmetros referentes ao ambiente podem ser vistos na tabela 4-15.

Tabela 4.15: Parâmetros do ambiente da simulação de cobertura móvel.

Parâmetros	Valores
Frequência de transmissão	1300 Hz
Potência Inicial	30 dB
Potência Máxima	40 dB
Altura efetiva da antena receptora	2 m
h_b – altura efetiva da antena transmissora [m]	30 m

Cada dinâmica de treinamento teve um tempo aproximado de 15 minutos de duração e será detalhada no próximo item.

4.3.4

Resultados – Simulação de Cobertura Móvel

Para acompanhar a evolução da otimização, foram obtidas as médias das avaliações das equipes durante o treinamento das redes e plotadas na figura 4-21.

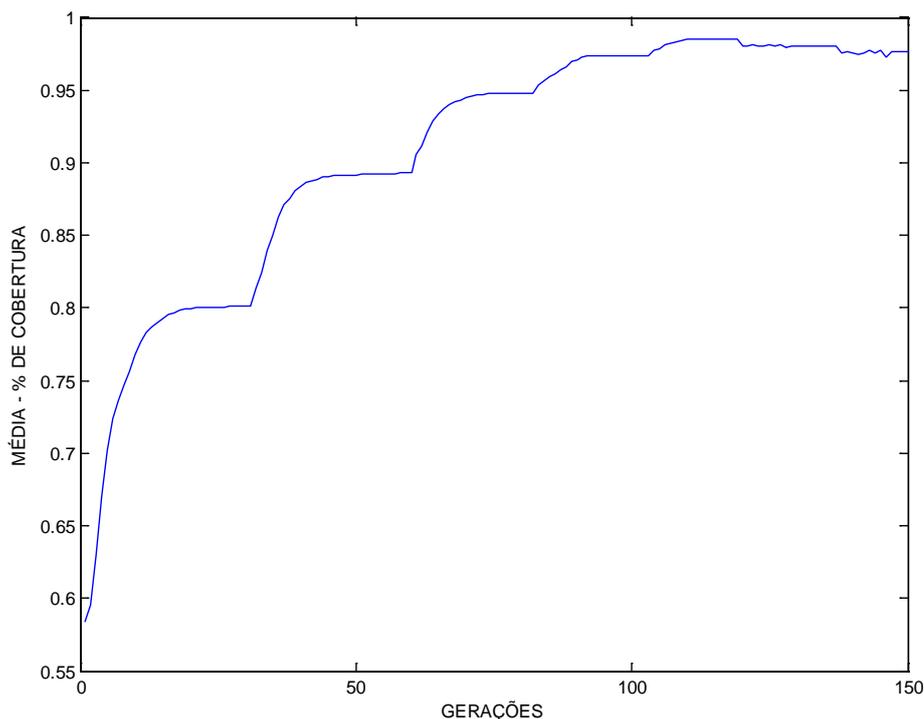


Figura 4.21: Evolução da avaliação média das equipes.

Pelo gráfico, nota-se uma rápida evolução nas primeiras 20 gerações, tendo o percentual de cobertura saltado de 61,45% para 80%, o que representa um aumento de 30,18% apenas neste início da evolução.

Com a estabilização das soluções, na 30ª geração o operador *lesion* foi ativado, alterando a estrutura das redes e proporcionando um novo incremento no percentual de cobertura, que saltou para 89,3% na geração de número 51.

Pode-se observar também que o operador *lesion* entra em ação ainda nas gerações de número 59, 81 e 100, permitindo o aumento da cobertura. Entretanto, como esperado, tais ganhos vão diminuindo conforme a aproximação da convergência.

Na 130ª geração, a média das avaliações atinge seu ponto máximo, no valor de 98,04%, indicando a convergência do algoritmo. Nas gerações de número 135 e 145 o operador *lesion* também atua, não conseguindo melhorar a avaliação, causando instabilidade e uma ligeira queda das avaliações ao final do processo.

O desempenho da melhor solução em cada geração pode ser analisado na figura 4-22 a seguir. No gráfico, pode-se notar também as estabilizações da evolução que ativam o operador *lesion* e a obtenção da melhor solução na 110ª geração, com 98,5% do percentual de cobertura. Ou seja, ao final do ajuste, apenas 30 dos 2000 móveis simulados não conseguiram se conectar a alguma

torre, o que significa um aumento de 60% em relação à configuração inicial.

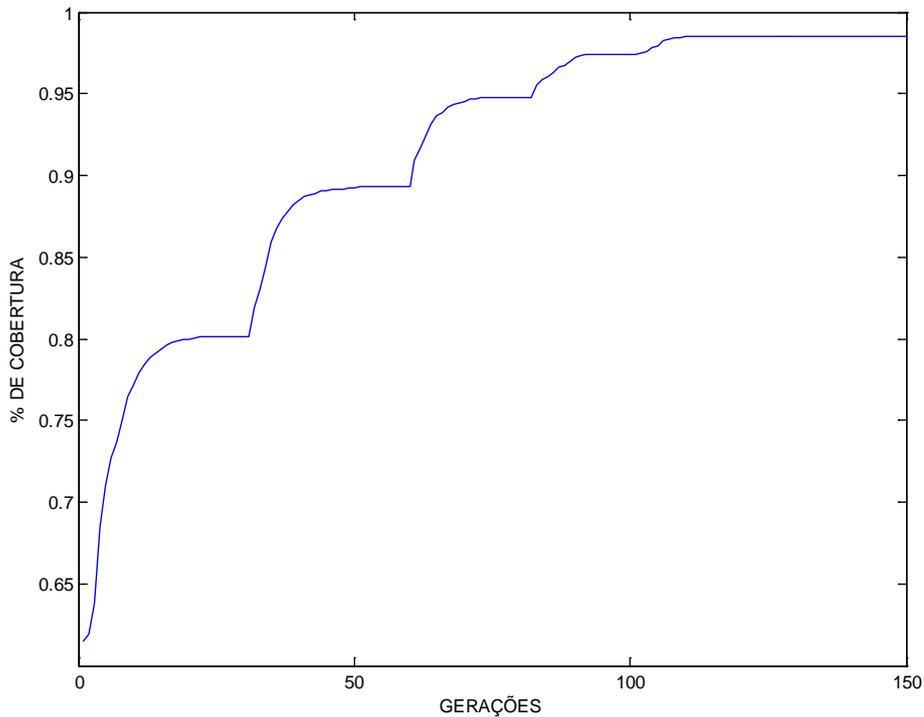


Figura 4.22: Evolução da melhor avaliação.

Apesar do foco do presente estudo ser o percentual de cobertura, uma análise também deve ser feita na potência disponível para a qualidade dos sinais. Para tal, foi feito um histograma com o percentual da potência disponível em cada antena que pode ser visto na figura 4-23.

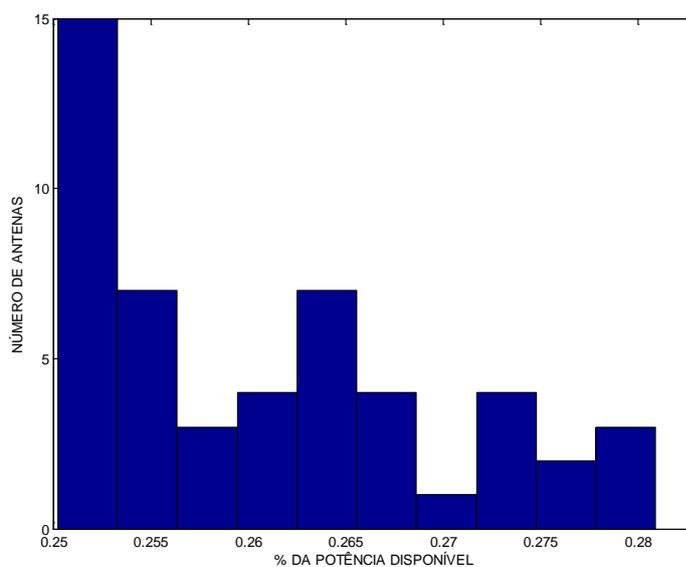


Figura 4.23: Percentual de potência disponível.

A faixa com maior frequência disponível foi a entre 25% e 25,6%, com 22 antenas operando nesse intervalo. Todas as outras antenas têm um percentual disponível acima de 26%, indicando que existe uma boa margem de potência disponível para a qualidade dos sinais.

Devido à particularidade da modelagem, a comparação com outros modelos propostos fica comprometida. Entretanto, uma simulação teste foi feita com um modelo que gerava e armazenava soluções aleatórias, em um novo mapeamento de torres e antenas. O resultado pode ser visto na figura 4-24.

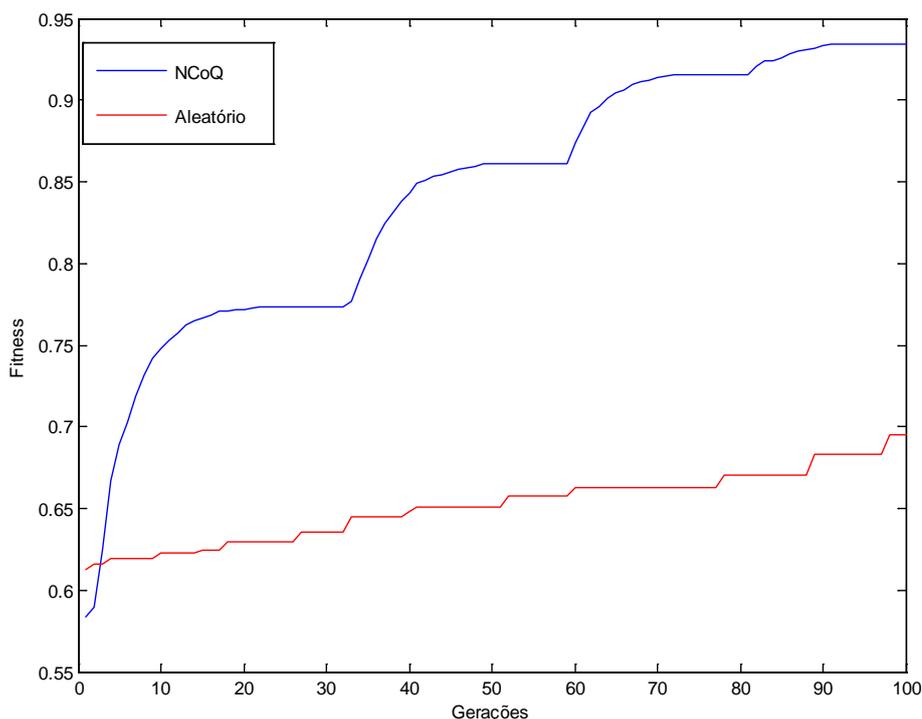


Figura 4.24: Comparação entre o modelo NCoQ e um modelo aleatório.

Nota-se, claramente, a diferença de desempenho entre as duas alternativas. Enquanto o modelo aleatório, mesmo partindo de soluções melhores, só atinge valores próximos a 70% de cobertura ao final do processo, o modelo NCoQ em menos de 10 gerações já apresentava resultados superiores, tendo atingido ao final da evolução, uma cobertura de 93,4%. No gráfico também fica bem demarcada as atuações do operador *lesion* após as estabilizações das soluções. Neste sentido, pode-se dizer que o modelo NCoQ apresentou resultados satisfatórios para a presente modelagem, atingindo, para o problema proposto, coberturas acima de 90%. Cabe ressaltar, que os ajustes são feitos de forma autônoma, o que valoriza ainda mais as soluções encontradas diante da complexidade da otimização.

Considerações Finais e Trabalhos Futuros

O desenvolvimento tecnológico tem possibilitado o aumento do uso de robôs em diversas situações do nosso cotidiano. Dessa maior interação homem-máquina surgem problemas cada vez mais complexos. Neste sentido, a busca por mecanismos mais efetivos de aprendizado por máquinas é fundamental, principalmente para a resolução de problemas onde o conhecimento do programador é limitado, como problemas em ambientes desconhecidos ou de grandes dimensões.

Aprendizado por reforço é uma das metodologias mais utilizadas para o aprendizado de máquinas. Nesta metodologia, o agente utiliza a recompensa recebida do ambiente para gerar sua política de ação. Porém, atribuir tal recompensa em um cenário com diversos agentes pode não ser uma tarefa trivial, além da alta dimensionalidade geralmente presente em problemas multi-agente.

O modelo NCoQ, principal contribuição deste trabalho, ao ser comparado com modelos de aprendizado por reforço multi-agente encontrados na literatura, apresenta como principal vantagem a simplificação na avaliação das soluções: enquanto os modelos de aprendizado por reforço exigem uma dinâmica de atribuição de recompensa aos agentes em cada etapa do processo de aprendizado, o modelo NCoQ exige apenas que a equipe de agentes cooperativos possa ser avaliada ao final do processo.

Em relação a outros modelos neuro-evolutivos, os estudos de casos feitos e apresentados neste estudo comprovaram o bom desempenho do modelo NCoQ, justificando as implementações realizadas, como a co-evolução quântica, o operador *lesion* multi-agente e os novos operadores de *crossover* e mutação quânticos.

O modelo proposto foi testado em três *benchmarks* distintos para validação. Inicialmente, uma modelagem no problema presa-predador foi feita visando a comparação do mesmo com outros modelos neuro-evolutivos, como o clássico e o precursor AEIQ-R, tendo o NCoQ apresentado o melhor desempenho na fase de testes.

Na segunda aplicação, um ambiente mais complexo foi modelado na aplicação *Multi-Rover*, cuja tarefa dos agentes era atuar de forma coordenada na exploração do ambiente coletando rubis através do uso de sensores, tendo o

modelo NCoQ apresentou um desempenho, em média, 28% melhor ao das metodologias comparadas.

Na última aplicação, que modelava a cobertura de operadoras de celular, o modelo NCoQ apresentou resultados satisfatórios, apresentando uma cobertura de aproximadamente 95% dos celulares. Para efeito de comparação, um modelo aleatório foi testado e, mais uma vez, a eficiência do modelo NCoQ foi constatada.

Diante dos resultados obtidos, pode-se afirmar que o modelo proposto atendeu de forma satisfatória aos desafios propostos, tendo grande capacidade de adaptação aos mais diversos tipos de ambientes aos quais foi submetido, sendo capaz de coordenar um grande número de agentes sem grandes complexidades na programação.

Apesar dos resultados satisfatórios obtidos em tarefas de alta complexidade, o modelo NCoQ carece de mais alguns ajustes. Outros mecanismos de atualização da população quântica podem ser utilizados, através de outros operadores quânticos. Também ainda há o limitante da arquitetura determinada pelo operador *lesion* ser repassada a todos os outros agentes, o que diminui a diversidade na população. Uma alternativa seria o uso da representação adotada pela neuro-evolução NEAT que permite o cruzamento entre genomas de dimensões distintas.

Outra carência do presente estudo é a falta de testes realizados em aplicações reais, já que todos os testes foram feitos em ambientes simulados. Por conta disso, é de se esperar uma queda de desempenho do modelo NCoQ em problemas robóticos práticos que, geralmente, apresentam dificuldades não tratadas neste estudo como, por exemplo, ruídos e erros de odometria. Algumas sugestões de aplicações deste tipo podem ser vistas adiante.

5.1

Futebol de Robôs

A proposta de uma partida de futebol entre robôs envolve características de extrema complexidade sob o ponto de vista da ciência robótica, envolvendo as partes físicas (*hardware*) e de programação dos robôs (*software*). Além disto, o ambiente é continuamente modificado em funções de fatores imprevisíveis, como a movimentação dos robôs do time adversário.

Como as regras proíbem a intervenção humana, a entrada de cada sistema é fornecida por imagens que são processadas em um computador central que retorna aos robôs, via wireless, os comandos para aquele instante de tempo.

Seria interessante, por conta da complexa dinâmica deste ambiente, verificar o funcionamento do algoritmo NCoQ para esta tarefa, já que o futebol necessita de estratégias cooperativas por parte dos jogadores de uma equipe.

5.2

Robôs Humanóides

Um robô humanóide é um robô cujo formato se assemelha ao corpo humano. Utilizam-se de atuadores para exercer as funções dos músculos e articulações e de sensores para receber informações e interagir com o ambiente.

Entretanto, não é uma tarefa simples obter a coordenação adequada entre esses atuadores nas diversas ações humanas como, por exemplo, num deslocamento bípede. Atuadores hidráulicos e elétricos têm um comportamento muito rígido e só podem agir de uma forma compatível com o uso de estratégias de controle de *feedback* relativamente complexas.

Modelos matemáticos são feitos para simularem estas situações. Na figura 5-1 podemos ver o modelo de mão artificial proposto por (Dragulesu & Ungureanu, 2008).

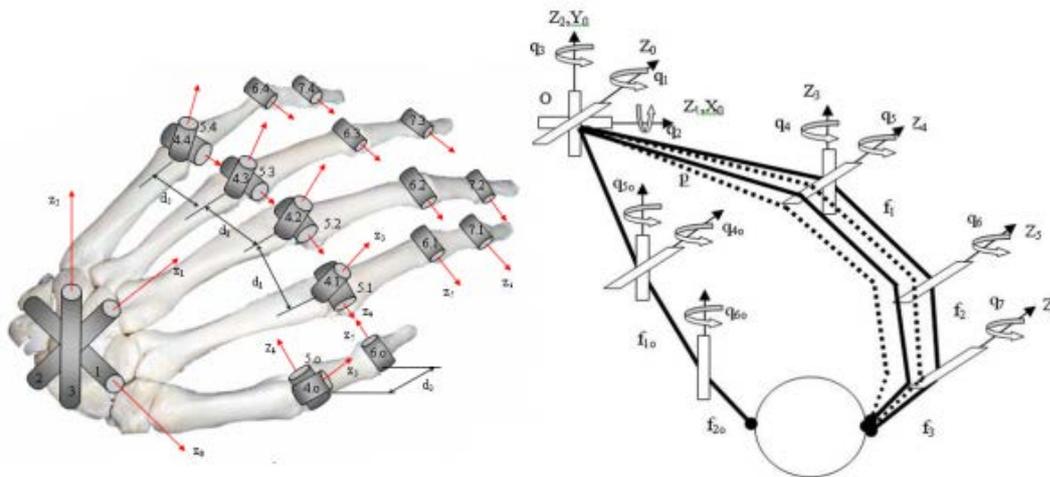


Figura 5-1 Modelo de mão artificial (Ungureanu, 2008).

Neste sentido, o modelo NCoQ pode ser utilizado nestes ambientes simulados para orientar as buscas por uma otimização adequada dos parâmetros do modelo, sendo as entradas das redes fornecidas pelos sensores e as saídas sendo os sinais de entrada dos atuadores.

Referências Bibliográficas

BIBLIOGRAPHY ABDESSALEM, Wahiba Ben; KARAA, Sami. MA-REM: A multi-agent system for renewable energy management. In: **Renewable Energy Congress (IREC), 2014 5th International**. IEEE, 2014. p. 1-6.

AGOGINO, Adrian. Design and control of large collections of learning agents. 2001. University of Texas at Austin.

ANATEL. (2015). <http://www.anatel.gov.br/dados/index.php/destaque-1/218-destaque-6>. Acesso em: 15 de out. 2015.

ARAGON-ZAVALA, Alejandro. **Antennas and propagation for wireless communication systems**. John Wiley & Sons, 2008.

ARENA, Paolo; PATANÉ, Luca; VITANZA, Alessandra. Autonomous learning of collaboration among robots. In: **The 2012 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2012. p. 1-8.

BARROS, Rodrigo Coelho et al. A survey of evolutionary algorithms for decision-tree induction. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 42, n. 3, p. 291-312, 2012.

BAZZAN, Ana Lúcia.. IA multiagente: mais inteligência, mais desafios. Jornada de Atualização em Informática (pp. 111-160). Rio de Janeiro: PUC-RIO, 2010.

BELEW, Richard K. Interposing an ontogenic model between genetic algorithms and neural networks. **Advances in Neural information processing systems**, p. 99-99, 1993.

BERNDT, Jan Ole; HERZOG, Otthein. Efficient multiagent coordination in dynamic environments. In: **Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on**. IEEE, 2011. p. 188-195.

BONABEAU, Eric; THERAULAZ, Guy; DENEUBOURG, Jean-Louis. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. **Proceedings of the Royal Society of London B: Biological Sciences**, v. 263, n. 1376, p. 1565-1569, 1996.

BUSE, D. P. et al. Agent-based substation automation. **IEEE Power and Energy Magazine**, v. 1, n. 2, p. 50-55, 2003.

BUSONI, Lucian; BABUSKA, Robert; DE SCHUTTER, Bart. A comprehensive survey of multiagent reinforcement learning. **IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews**, **38 (2)**, 2008.

CALDERONE, Nicholas W.; PAGE JR, Robert E. Genotypic variability in age polyethism and task specialization in the honey bee, *Apis mellifera* (Hymenoptera: Apidae). **Behavioral Ecology and Sociobiology**, v. 22, n. 1, p. 17-25, 1988.

CHAINBI, Walid; HANACHI, Chihab; SIBERTIN-BLANC, Christophe. The multi-agent prey/predator problem: A Petri net solution. In: **Proceedings of the Symposium on Discrete Events and Manufacturing systems, Lille, France**. 1996.

CHANDRA, Rohitash. Competitive two-island cooperative coevolution for training Elman recurrent networks for time series prediction. In: **2014 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2014. p. 565-572.

COLBY, Mitchell; KNUDSON, Matthew D.; TUMER, Kagan. Multiagent flight control in dynamic environments with cooperative coevolutionary algorithms. *AAAI Spring Symposium Series*, (pp. 110 - 1150), 2014.

CORRÊA, Marcelo França. **Modelos Neuro-Fuzzy Hierárquicos com Aprendizado por Reforço para Multi-Agentes Inteligentes**. 2011. Tese de Doutorado. PUC-Rio.

CORRÊA, Marcelo França; VELLASCO, Marley; FIGUEIREDO, Karla. Multi-agent systems with reinforcement hierarchical neuro-fuzzy models. **Autonomous Agents and Multi-Agent Systems**, v. 28, n. 6, p. 867-895, 2014.

CRUZ, André Vargas Abs da. **Algoritmos evolutivos com inspiração quântica para problemas com representação numérica**. 2007. Tese de Doutorado. PUC-Rio.

CRUZ, André Vargas Abs da; VELLASCO, Marley MBR; PACHECO, Marco Aurélio C. Quantum-Inspired Evolutionary Algorithms applied to numerical optimization problems. In: **IEEE Congress on Evolutionary Computation**. IEEE, 2010. p. 1-6.

DAVIDSON, Euan M. et al. Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data. **IEEE Transactions On Power Systems Pwrs**, v. 21, n. 2, p. 559, 2006.

DIAS, Douglas Mota. **Programação Genética Linear com Inspiração Quântica**. 2010. Tese de Doutorado. PUC-Rio.

DRAGULESCU, Doina; UNGUREANU, Loredana. The modeling process of a human hand prosthesis. In: **2007 4th International Symposium on Applied Computational Intelligence and Informatics**. IEEE, 2007. p. 263-268.

EL-MESALLAMY, Mostafa et al. Coordination of directional overcurrent relays using Artificial Bee Colony. In: **Electricity Distribution (CIRED 2013), 22nd International Conference and Exhibition on**. IET, 2013. p. 1-4.

FLOREANO, Dario; URZELAI, Joseba. Evolutionary robots with on-line self-organization and behavioral fitness. **Neural Networks**, v. 13, n. 4, p. 431-443, 2000.

FROZZA, Rejane. **Crítérios para avaliação de coordenação multiagente**. 2004. Tese de Doutorado. Universidade Federal do Rio Grande do Sul.

GAO, Shuli; GIVIGI, Sidney N.; BEAULIEU, Alain JG. FPGA implementation of multiple Pursuit-Evasion games with decentralized Learning Automata. In: **Systems Conference (SysCon), 2014 8th Annual IEEE**. IEEE, 2014. p. 78-82.

GODOY, Arlida Schmidt. Introdução à pesquisa qualitativa e suas possibilidades. **Revista de administração de empresas**, v. 35, n. 2, p. 57-63, 1995.

GOMEZ, Faustino; MIIKKULAINEN, Risto. Incremental evolution of complex general behavior. **Adaptive Behavior**, v. 5, n. 3-4, p. 317-342, 1997.

GOMEZ, Faustino John; MIIKKULAINEN, Risto. **Robust non-linear control through neuroevolution**. Computer Science Department, University of Texas at Austin, 2003.

GÓMEZ, Iván; FRANCO, Leonardo; JEREZ, José M. Neural network architecture selection: can function complexity help?. **Neural Processing Letters**, v. 30, n. 2, p. 71-87, 2009.

GOODWINE, Bill; ANTSAKLIS, Panos. Multiagent coordination exploiting system symmetries. In: **Proceedings of the 2010 American Control Conference**. IEEE, 2010. p. 830-835.

GRUAU, Frédéric; WHITLEY, Darrell; PYEATT, Larry. A comparison between cellular encoding and direct encoding for genetic neural networks. In: **Proceedings of the 1st annual conference on genetic programming**. MIT Press, 1996. p. 81-89.

GUESTIN, Carlos; LAGOUDAKIS, Michail; PARR, Ronald. Coordinated reinforcement learning. In: **ICML**. 2002. p. 227-234.

HAN, Kuk-Hyun; KIM, Jong-Hwan. Genetic quantum algorithm and its application to combinatorial optimization problem. In: **Evolutionary Computation, 2000. Proceedings of the 2000 Congress on**. IEEE, 2000. p. 1354-1360.

HAN, Kuk-Hyun; KIM, Jong-Hwan. Quantum-inspired evolutionary algorithms with a new termination criterion, H & epsilon; gate, and two-phase scheme. **IEEE transactions on evolutionary computation**, v. 8, n. 2, p. 156-169, 2004.

HATA, Masaharu. Empirical formula for propagation loss in land mobile radio services. **IEEE transactions on Vehicular Technology**, v. 29, n. 3, p. 317-325, 1980.

HAYKIN, Simon. **Neural Networks: a comprehensive foundation**. v. 2, Prentice Hall, 2004.

HU, Yujing; GAO, Yang; AN, Bo. Multiagent reinforcement learning with unshared value functions. **IEEE transactions on cybernetics**, v. 45, n. 4, p. 647-662, 2015.

HWANG, Kao-Shing; CHEN, Yu-Jen; JIANG, Wei-Cheng. Learning acceleration by policy sharing. In: **Intelligent Control and Automation (WCICA), 2011 9th World Congress on**. IEEE, 2011. p. 725-729.

JAMIL, Momin; YANG, Xin-She. A literature survey of benchmark functions for global optimisation problems. **International Journal of Mathematical Modelling and Numerical Optimisation**, v. 4, n. 2, p. 150-194, 2013.

JANG, Jun-Su; HAN, Kuk-Hyun; KIM, Jong-Hwan. Face detection using quantum-inspired evolutionary algorithm. In: **Evolutionary Computation, 2004. CEC2004. Congress on**. IEEE, 2004. p. 2100-2106.

JENNINGS, Nick R. Coordination techniques for distributed artificial intelligence. **Foundations of distributed artificial intelligence**, p. 187-210, 1996.

JÚNIOR, Paulo Roberto Ferreira. **Coordenação de sistemas multiagente atuando em cenários complexos: uma abordagem baseada na divisão de trabalho dos insetos sociais**. 2008. Tese de Doutorado. Universidade Federal do Rio Grande do Sul.

KENNEDY, J.; EBERHART, C. Particle swarm optimization: proceeding of IEEE International Conference on Neural Networks. **Piscataway, NJ: IEEE, c1995**, v. 1948, 1942.

KOK, Jelle R. et al. Utile coordination: Learning interdependencies among cooperative agents. In: **EEE Symp. on Computational Intelligence and Games, Colchester, Essex**. 2005. p. 29-36.

KOMURA, Keigo; MUTOH, Atsuko; INUZUKA, Nobuhiro. Application of Inductive Logic Programming to Produce Emergent Behavior in an Artificial Society. In: **Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference on**. IEEE, 2014. p. 942-947.

KOŠINÁR, Michal; KOHUT, Ondrej. Applying Statistical Methods in KM of a Multiagent System. **Journal Compilation Institute of Philosophy**. 2010. p. 201-216.

LAU, Nuno; REIS, Luis Paulo. **FC Portugal-high-level coordination methodologies in soccer robotics**. INTECH Open Access Publisher, 2007.

LINGHU, Bin; CHEN, Feng. An Intelligent Multi-agent Approach for Flood Disaster Forecasting Utilizing Case Based Reasoning. In: **Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on**. IEEE, 2014. p. 182-185.

LIU, Jing; ZHONG, Weicai; JIAO, Licheng. A multiagent evolutionary algorithm for combinatorial optimization problems. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 40, n. 1, p. 229-240, 2010.

LOWELL, Jessica; GRABKOVSKY, Sergey; BIRGER, Kir. Comparison of NEAT and HyperNEAT performance on a strategic decision-making problem. In: **Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on**. IEEE, 2011. p. 102-105.

MALONE, Thomas W.; CROWSTON, Kevin. The interdisciplinary study of coordination. **ACM Computing Surveys (CSUR)**, v. 26, n. 1, p. 87-119, 1994.

MCARTHUR, Stephen DJ; STRACHAN, Scot M.; JAHN, Gordon. The design of a multi-agent transformer condition monitoring system. **IEEE Transactions on Power Systems**, v. 19, n. 4, p. 1845-1852, 2004.

MENDOZA, Leonardo Forero; VELLASCO, Marley; FIGUEIREDO, Karla. Intelligent multiagent coordination based on reinforcement hierarchical neuro-fuzzy models. **International journal of neural systems**, v. 24, n. 08, p. 1450031, 2014.

MENG, Xiangping et al. An improvement to the coordination method of ant colony algorithm. In: **Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on**. IEEE, 2012. p. 114-117.

MICHALEWICZ, Zbigniew. Genetic algorithms data structures evolution programs. **Springer, Berlin**, 1996.

MIKKULAINEN, Risto et al. Multiagent learning through neuroevolution. In: **IEEE World Congress on Computational Intelligence**. Springer Berlin Heidelberg, 2012. p. 24-46.

MOHAMMED, Amgad M. et al. Quantum crossover based quantum genetic algorithm for solving non-linear programming. In: **Informatics and Systems (INFOS), 2012 8th International Conference on**. IEEE, 2012. p. BIO-145-BIO-153.

MORE, Surekha; BHARAMBE, Ujwala. Intelligent web mining technique using evolutionary algorithms. In: **Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on**. IEEE, 2014. p. 508-514.

MORIARTY, David Eric. **Symbiotic evolution of neural networks in sequential decision tasks**. 1997. Tese de Doutorado. University of Texas at Austin.

MORIARTY, David E.; MIKKULAINEN, Risto. Efficient reinforcement learning through symbiotic evolution. **Machine learning**, v. 22, n. 1-3, p. 11-32, 1996.

NAGATA, Takeshi; SASAKI, Hiroshi. A multi-agent approach to power system restoration. **IEEE Transactions on power systems**, v. 17, n. 2, p. 457-462, 2002.

NITSCHKE, Geoff. Neuro-evolution methods for gathering and collective construction. In: **European Conference on Artificial Life**. Springer Berlin Heidelberg, 2009. p. 115-123.

NITSCHKE, G. **Neuro-evolution for emergent specialization in collective behavior systems**. 2008. Tese de Doutorado. Vrije Universiteit Amsterdam.

NOIROT, C. H.; PASTEELS, J. M. Ontogenetic development and evolution of the worker caste in termites. **Experientia**, v. 43, n. 8, p. 851-860, 1987.

OKUMURA, Y.; OHMORI, Et; STRENGTH, Field. its variability in VHF and UHF land mobile Service', Rev. **Elec. Com. Lab**, v. 16, p. 825-873, 1968.

OSSOWSKI, Sascha; GARCÍA-SERRANO, Ana. A knowledge-level model of coordination. In: **Australian Workshop on Distributed Artificial Intelligence**. Springer Berlin Heidelberg, 1995. p. 46-57.

ÖZVEREN, Cüneyt Süheyl; SAPELUK, A. T.; BIRCH, Alan. An investigation into using neuro-evolution of Augmenting Topologies (NEAT) for short term load forecasting (STFL). In: **Power Engineering Conference (UPEC), 2014 49th International Universities**. IEEE, 2014. p. 1-5.

PANAIT, Liviu; LUKE, Sean. Cooperative multi-agent learning: The state of the art. **Autonomous agents and multi-agent systems**, v. 11, n. 3, p. 387-434, 2005.

PANAIT, Liviu. **The analysis and design of concurrent learning algorithms for cooperative multiagent systems**. 2006. Tese de Doutorado. George Mason University.

PARSONS, J. Thomas et al. Focal adhesion kinase: a regulator of focal adhesion dynamics and cell movement. **Oncogene**, v. 19, n. 49, p. 5606-5613, 2000.

PINHO, Anderson Guimarães de. **Algoritmo Evolucionário com Inspiração Quântica e Representação Mista Aplicado a Neuroevolução**. 2010. Tese de Doutorado. PUC-Rio.

PIQUER, Alvaro Garcia et al. Large-scale experimental evaluation of cluster representations for multiobjective evolutionary clustering. **IEEE transactions on evolutionary computation**, v. 18, n. 1, p. 36-53, 2014.

POTTER, Mitchell A.; DE JONG, Kenneth A. Cooperative coevolution: An architecture for evolving coadapted subcomponents. **Evolutionary computation**, v. 8, n. 1, p. 1-29, 2000.

PUROHIT, Seema; MANTRI, Shruti. Game theoretic multi-agent approach to traffic flow control. In: **Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on**. IEEE, 2013. p. 1902-1905.

QUIÑONEZ, Yadira; TOSTADO, Iván; SÁNCHEZ, Omar. Coordination Model for Multi-robot Systems Based on Cooperative Behaviors. In: **Artificial Intelligence (MICA), 2013 12th Mexican International Conference on**. IEEE, 2013. p. 33-37.

REGO, Renata do et al. Evolutionary Algorithm for 3D Object Reconstruction from Images. In: **2006 Ninth Brazilian Symposium on Neural Networks (SBRN'06)**. IEEE, 2006. p. 54-59.

REIS, Luís Paulo Gonçalves dos et al. **Coordenação em sistemas multi-agente: Aplicações na gestão universitária e futebol robótico**. 2012. Tese de Doutorado. Universidade do Porto

RIDLEY, Mark. **Evolution**. Malden: Oxford Readers, 2004.

RUSSELL, Stuart; NORVIG, Peter; **Artificial Intelligence: A modern approach**. **Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs**, v. 25, 1995.

SALICHON, Max; TUMER, Kagan. Evolving a multiagent controller for micro aerial vehicles. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 42, n. 6, p. 1772-1783, 2012.

SCHRUM, Jacob; MIIKKULAINEN, Risto. Evolving multimodal networks for multitask games. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 4, n. 2, p. 94-111, 2012.

SELIGMANN, Hervé. Resource partition history and evolutionary specialization of subunits in complex systems. **Biosystems**, v. 51, n. 1, p. 31-39, 1999.

SELLNER, Brennan et al. Coordinated multiagent teams and sliding autonomy for large-scale assembly. **Proceedings of the IEEE**, v. 94, n. 7, p. 1425-1444, 2006.

SHENG, Zhang; WANLU, Jiang. A novel quantum genetic algorithm and its application. In: **Natural Computation (ICNC), 2012 Eighth International Conference on**. IEEE, 2012. p. 613-617.

SHI, Peng; SHEN, Qikun. Cooperative control of multi-agent systems with unknown state-dependent controlling effects. **IEEE Transactions on Automation Science and Engineering**, v. 12, n. 3, p. 827-834, 2015.

SOHANGIR, Soroosh; RAHIMI, Shahram; GUPTA, Bidyut. Optimized feature selection using NeuroEvolution of Augmenting Topologies (NEAT). In: **IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint**. IEEE, 2013. p. 80-85.

STANLEY, K. O. **Efficient Evolution of Neural Networks through Complexification**. 2004. Tese de Doutorado. Artificial Intelligence Laboratory. The University of Texas at Austin, Austin, USA (August 2004).

STONE, Peter; VELOSO, Manuela. Multiagent systems: A survey from a machine learning perspective. **Autonomous Robots**, v. 8, n. 3, p. 345-383, 2000.

SU, Haitao et al. Validity Evaluation of Team Learning Based on DEA. In: **Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on**. IEEE, 2009. p. 939-942.

TIBERMACHINE, Ahmed; DJEDI, NourEddine. NEAT neural networks to control and simulate virtual creature's locomotion. In: **Multimedia Computing and Systems (ICMCS), 2014 International Conference on**. IEEE, 2014. p. 9-14.

TONDA, Alberto; LUTTON, Evelyne; SQUILLERO, Giovanni. A benchmark for cooperative coevolution. **Memetic Computing**, v. 4, n. 4, p. 263-277, 2012.

VAKILI, Shervin et al. Particle Swarm Optimization for Run-Time Task Decomposition and Scheduling in Evolvable MPSoC. In: **Computer Engineering and Technology, 2009. ICCET'09. International Conference on**. IEEE, 2009. p. 28-32.

VALKEALAHTI, Kimmo et al. WCDMA common pilot power control for load and coverage balancing. In: **Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on**. IEEE, 2002. p. 1412-1416.

VASSILIADES, Vassilis; CLEANTHOUS, Aristodemos; CHRISTODOULOU, Chris. Multiagent reinforcement learning: spiking and nonspiking agents in the iterated prisoner's dilemma. **IEEE Transactions on Neural Networks**, v. 22, n. 4, p. 639-653, 2011.

WANG, Haixia; LIU, Fengjun. The impact of team external learning on team effectiveness. In: **2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering**. 2013.

WEI, Fei; WANG, Yuping; HUO, Yuanliang. Smoothing and auxiliary functions based cooperative coevolution for global optimization. In: **2013 IEEE Congress on Evolutionary Computation**. IEEE, 2013. p. 2736-2741.

WEISS, Gerhard. **Multiagent systems: a modern approach to distributed artificial intelligence**. MIT press, 1999.

WENSELEERS, Tom; RATNIEKS, Francis LW; BILLEN, Johan. Caste fate conflict in swarm-founding social Hymenoptera: an inclusive fitness analysis. **Journal of evolutionary biology**, v. 16, n. 4, p. 647-658, 2003.

WERKEMA, Maria Cristina Catarino. **Ferramentas estatísticas básicas para o gerenciamento de processos**. Universidade Federal de Minas Gerais. Escola de Engenharia. Fundação Christiano Ottoni, 1995.

WHITLEY, Darrell et al. Genetic reinforcement learning for neurocontrol problems. In: **Genetic Algorithms for Machine Learning**. Springer US, 1993. p. 103-128.

WOOLDRIDGE, Michael. **An introduction to multiagent systems**. John Wiley & Sons, 2009.

WOOLDRIDGE, Michael; JENNINGS, Nicholas R. Agent theories, architectures, and languages: a survey. In: **International Workshop on Agent Theories, Architectures, and Languages**. Springer Berlin Heidelberg, 1994. p. 1-39.

WOOLDRIDGE, Michael. Intelligent Agents In G. Weiss, editor: Multiagent systems. 1999.

WOOLDRIDGE, Michael; JENNINGS, Nicholas R. Intelligent agents: Theory and practice. **The knowledge engineering review**, v. 10, n. 02, p. 115-152, 1995.

YAO, Xin; LIU, Yong. A new evolutionary system for evolving artificial neural networks. **IEEE transactions on neural networks**, v. 8, n. 3, p. 694-713, 1997.

YONG, Chern Han; MIIKKULAINEN, Risto. **Cooperative coevolution of multi-agent systems**. University of Texas at Austin, Austin, TX, 2001.

ZEBULUM, Ricardo Salem; PACHECO, Marco Aurélio; VELLASCO, Marley Maria Be. **Evolutionary electronics: automatic design of electronic circuits and systems by genetic algorithms**. CRC press, 2001.

ZHANG, Zhen; ZHAO, Dongbin. Cooperative multiagent reinforcement learning using factor graphs. In: **Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on**. IEEE, 2013. p. 797-802.