RICHARDSON RIBEIRO

ANÁLISE DO IMPACTO DA TEORIA DAS REDES SOCIAIS EM TÉCNICAS DE OTIMIZAÇÃO E APRENDIZAGEM MULTIAGENTE BASEADAS EM RECOMPENSAS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito para obtenção do título de Doutor em Informática.

CURITIBA

2010

RICHARDSON RIBEIRO

ANÁLISE DO IMPACTO DA TEORIA DAS REDES SOCIAIS EM TÉCNICAS DE OTIMIZAÇÃO E APRENDIZAGEM MULTIAGENTE BASEADAS EM RECOMPENSAS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito para obtenção do título de Doutor em Informática.

Área de Concentração: *Agentes de Software*

Orientador: Prof. Dr. Fabrício Enembreck

CURITIBA

2010

Agradecimentos

Este trabalho somente poderia ter sido terminado com ajuda de várias pessoas. Primeiramente, meu agradecimento de forma especial para o orientador desde trabalho, Prof. Dr. Fabrício Enembreck. Prof. Dr. Fabrício não somente estimulou continuamente minhas pesquisas com valorosas orientações técnicas e rigores científicos, mas também forneceu acesso aos equipamentos laboratoriais, materiais didáticos e suporte as viagens de pesquisas. Eu também gostaria de agradecê-lo por me confiar suas pesquisas, oportunizando atuar em seu grupo de estudo onde pude interagir com demais pesquisadores e alunos. Obrigado pela amizade, respeito e profissionalismo, que me fazem acreditar na continuidade de novas descobertas e pesquisas. Foram enormes suas contribuições, lapidando minha formação acadêmica e profissional.

Meus respeitos e agradecimentos aos demais professores do laboratório de Agentes de Software do PPGIA/PUCPR. Aos líderes do grupo de pesquisa Prof. Dr. Bráulio C. Ávila e Prof. Dr. Edson E. Scalabrin, que oportunizam estudos, pesquisas e financiamentos aos acadêmicos com seus projetos inovadores. Obrigado Ávila e Scalabrin pelas sugestões e críticas, que ajudaram a entender a importância da objetividade.

Obrigado aos professores Dr. Gustavo A. G. Lugo (UTFPR), Dr. Milton P. Ramos (TECPAR) e Dr. Júlio C. Nievola (PPGIA) que colaboraram com discussões e direcionamentos desde trabalho. Agradeço também ao prof. Dr. Alessandro L. Koerich (PPGIA) nas redações dos artigos.

Aos amigos de estudos e laboratório: Allan, André, Bruno, Marcos, Osmar, entre outros, no qual juntos compartilhamos conhecimentos, alegrias e desesperos. Obrigado André pela moradia e demais favores, na qual sempre ficará minha gratidão.

Ao meu pai (*in memoriam*) que esteve presente nos primeiros dois anos da tese. Obrigado pelos conselhos, jamais alcançarei sua cultura e seu conhecimento, meu exemplo de vida e superação, sempre na lembrança. A minha mãe, que sempre cuidou bem de mim com muito amor e carinho, obrigado. Ao meu irmão Charlison, obrigado pela atenção e favores prestados. Vocês têm grande significado na minha vida.

A minha esposa Adriana, a pessoa mais motivadora para a realização desse trabalho. Essa tese é especialmente dedicada a ela, sempre carinhosa e me fazendo acreditar que valeria a pena todo o trabalho.

Agradeço aos professores Orestes, Gerson, Adriano e Douglas, pelas discussões e auxílios técnicos. Prof. Emílio Evers Neto, coordenador do curso de Administração e Prof. José Alceu Valério, reitor da Universidade do Contestado, que me oportunizaram nesses anos entender a estrutura e as dimensões das instituições de ensino. Obrigado pelo ambiente de trabalho, e que a Comissão Própria de Avaliação (CPA) continue contribuindo nos processos institucionais.

Sem esquecer ainda de uma pessoa importante, que considero minha segunda mãe, Maria da Conceição, que me mostrou como a generosidade e o afeto pode aquecer e amparar nos momentos mais difíceis. Muito obrigado.

Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a PUCPR pelo apoio financeiro em forma de bolsa de estudos.

Por fim, muito obrigado a todos da banca avaliadora que contribuíram com críticas e sugestões para as melhorias deste trabalho.

Resumo

Este trabalho é dedicado ao estudo da aplicação das teorias sociais para a construção de estruturas de relacionamento capazes de influenciar comportamentos locais gerados a partir de recompensas em indivíduos de um sistema multiagente. A interação entre modelos de coordenação com a teoria das redes sociais no processo da tomada de decisão gera uma estrutura social à medida que as interações dos indivíduos ocorrem com as recompensas geradas. Técnicas de otimização por enxames e aprendizagem por reforço são baseadas em recompensas e geralmente são utilizadas para melhorar o comportamento e a coordenação dos indivíduos de um sistema. É possível observar com esses princípios que a sociabilidade dos agentes influencia nas atividades em comum, devido às atitudes comportamentais que estão relacionadas a teorias de ação, modelos de sistemas sociais, sistemas multiagente e teoria de redes sociais, que descrevem o impacto das relações observadas na rede formada pelos agentes. Neste contexto, é estudado neste trabalho como a sociabilidade dos agentes pode contribuir para melhorar e adaptar métodos de coordenação com a análise de redes sociais, alterando as recompensas geradas com algoritmos de aprendizagem por reforço, promovendo a convergência do sistema e a qualidade das políticas no processo de aprendizagem. Os métodos são testados em problemas de otimização combinatória, permitindo avaliar a vantagem dos aspectos que afetam o desempenho da abordagem proposta, como (i) a quantidade de agentes no ambiente; (ii) os parâmetros de aprendizagem; (iii) a qualidade da política; (iv) o compartilhamento de recompensas; e (v) a estrutura social. Resultados mostram que a identificação de comportamentos sociais e a estrutura social construída com a interação dos indivíduos contribuem significativamente para a melhoria do processo de coordenação.

Palavras-Chave: Análise de redes sociais, otimização por colônia de formigas, aprendizagem por reforço, coordenação e sistemas multiagente.

Abstract

This work is dedicated to study the application of the social theories to the construction of relationships structures able to influence local behaviors generated from individuals' rewards of a multiagent system. The interaction between coordination models with the social networks theory in decision-making process generates a social structure as individuals' interactions occur with generated rewards. Swarm optimization techniques and reinforcement learning are based in rewards and are usually used to improve the individuals' behavior and coordination of a system. It is possible to observe with these principles that the sociability of the agents influence in activities in common, due to behavioral attitudes that are related to action theories, social systems models, multiagent systems and social network theory, which describe the impact observed in the relationships network generated. In this context, it is studied in this work how the sociability contribute to improve and adapt coordination methods with social network analysis, changing the rewards generated with reinforcement algorithms, fostering the convergence of the system and improving policies quality. The methods have been evaluated in combinatorial optimization problems, allowing evaluate the impact of the following aspects that affect the performance of the proposed approach: (i) the amount of agents in the environment, (ii) the learning parameters (iii) the quality of policies, (iv) the shared rewards, (v) the social structure, and (vi) the benefits achieved. Results show that identifying social behaviors and social structure generated from individuals' behavior, the coordination process improves significantly.

Keywords: Social networks analysis, ant colony optimization, reinforcement learning, coordination and multiagent systems.

Sumário

CAPÍTULO 1 16

INTRODUÇÃO 16

1.1 PROBLEMA 18

1.2 HIPÓTESES 19

1.3 OBJETIVOS 20

1.4 ORGANIZAÇÃO DO TRABALHO 20

CAPÍTULO 2 22

APRENDIZAGEM E COORDENAÇÃO EM SISTEMAS MULTIAGENTE 22

2.1 AGENTES INTELIGENTES 22

2.2 COORDENAÇÃO DOS AGENTES 25

2.3 MÉTODOS DE COORDENAÇÃO E APRENDIZAGEM PARA SISTEMAS MULTIAGENTE 32

2.3.1 Coordenação por Interação 32

2.3.2 Coordenação por Sincronização 35

2.3.3 Coordenação por Planejamento 37

2.3.4 Coordenação Reativa 41

2.3.5 Coordenação por Formação de Coalizão 44

2.3.6 Otimização Distribuída de Restrição para Coordenação de Sistemas Multiagente 46

2.4 CRITÉRIOS DE ANÁLISE E COMPARAÇÃO PARA COORDENAÇÃO 49

2.5 CONSIDERAÇÕES FINAIS 54

CAPÍTULO 3 55

TEORIA DAS REDES SOCIAIS 55

3.1 DEFINIÇÕES DE REDES SOCIAIS 56

3.1.1 Classificação das Redes Sociais 57

3.2 FUNDAMENTOS MATEMÁTICOS E A TEORIA DOS GRAFOS 60

3.2.1 Ciclos Hamiltonianos 63

3.2.2 Teoria dos Grafos na Análise de Redes Sociais 63

3.3 ABORDAGENS COMPUTACIONAIS 69

3.4 REDES SOCIAIS E SISTEMAS MULTIAGENTE 70
3.5 CONSIDERAÇÕES FINAIS 71
CAPÍTULO 4 73
APRENDIZAGEM POR REFORÇO E OTIMIZAÇÃO POR ENXAMES 73
4.1 DEFINIÇÕES DA APRENDIZAGEM POR REFORÇO 74
4.1.1 Características da Aprendizagem por Reforço 75
4.1.2 Elementos Fundamentais da Aprendizagem por Reforço 76
4.1.3 Processos Markovianos 78
4.2 ALGORITMOS DE APRENDIZAGEM POR REFORÇO 80
4.2.1 Algoritmo Q-learning 80
4.2.2 Algoritmo R-learning 82
4.2.3 Algoritmo H-learning 83
4.2.4 Algoritmo $Q(\lambda)$ 84
4.2.5 Algoritmo Sarsa 85
4.2.6 Algoritmo Dyna 86
4.3 INTELIGÊNCIA BASEADA EM ENXAMES 86
4.3.1 Otimização por Enxames de Partículas 87
4.3.2 Inteligência Baseada em Cardume de Peixes 88
4.3.3 Otimização por Colônia de Formigas 89
4.4 ALGORITMOS BASEADOS EM COLÔNIA DE FORMIGAS 91
4.4.1 Ant System 92
4.4.2 Ant Colony System 93
4.4.3 Ant-Q 95
4.4.4 Fast Ant System 97
4.4.5 Antabu 98
4.4.6 AS-rank 98
4.4.7 Resoluções com Algoritmos de Colônia de Formigas 99
4.5 CONSIDERAÇÕES FINAIS 104
CAPÍTULO 5 105
ENFOQUE PROPOSTO 105
5.1 IMPACTO DAS RECOMPENSAS EM APRENDIZAGEM POR REFORÇO 106
5.1.2 Aprendizagem por Recompensas Partilhadas 107

5.1.3 Modelos de Compartilhamento de Recompensas para Aprendizagem Multiagente 111
5.1.4 Modelo Híbrido de Aprendizagem 116
5.1.5 Modelo Híbrido vs. Modelos Contínuo, Discreto e Dirigido por Objetivo 118
5.2 ANÁLISE DO ANT-Q 122
5.2.1 Resultados Experimentais 126
5.2.2 Estratégias de Atualização de Políticas para Ambientes Dinâmicos 131
5.3 SANT-Q (SOCIAL ANT-Q): UM ALGORITMO DE OTIMIZAÇÃO BASEADO EM COLÔNIA DE FORMIGAS, APRENDIZAGEM POR REFORÇO E TEORIAS SOCIAIS 140
5.3.1 Redes Baseadas em Relações 142
5.3.2 Construção da Rede de Relacionamentos com o SAnt-Q (Social Ant-Q) .. 145
5.3.3 Resultados Experimentais 155
5.3.4 Método de Otimização Social 166
CAPÍTULO 6 176
CONCLUSÕES E DISCUSSÕES FINAIS 176
6.1 TRABALHOS FUTUROS 178
6.2 PUBLICAÇÕES RELACIONADAS 179
REFERÊNCIAS BIBLIOGRÁFICAS 181

Lista de Figuras

Figura 2.1: Modelo abstrato de agentes inseridos em um ambiente 24
Figura 2.2: Tipos de relações entre ações (Ferber, 1999) 27
Figura 2.3: Classificação da coordenação (Moulin e Chaib-Draa, 1996) 28
Figura 2.4: Sincronização de ações (Ferber, 1999) 36
Figura 2.5: Divisão tradicional do sistema de controle em módulos funcionais (Brooks, 1990) 41
Figura 2.6: Divisão do sistema em camadas de tarefas (Brooks, 1990) 42
Figura 2.7: O sistema pode ser particionado em qualquer nível, e as camadas abaixo formam um completo sistema de controle (Brooks, 1990) 42
Figura 2.8: Diagrama do eco-agente (Ferber, 1999) 44
Figura 2.9: Pseudo-árvore gerada a partir de um grafo de restrições (Modi *et al.* 2005) 47
Figura 3.1: Redes do mundo pequeno, onde: a) rede sem ligações *shortcut*; b) rede com poucas *shortcut*; e c) mundo pequeno com muitos *shortcuts*, semelhante a um grafo quase completo (Gaston e DesJardins, 2005) 58
Figura 3.2: V = $\{v_1, v_2, v_3, v_4, v_5\}$ e E = $\{v_1\ v_2, v_1\ v_3, v_2\ v_4, v_3\ v_4, v_1\ v_1\}$ 60
Figura 3.3: As arestas dos vértices *v* e *w* são paralelas 60
Figura 3.4: Grafos isomorfos 61
Figura 3.5: Exemplo de subgrafo 62
Figura 3.6: Sociograma formado pelas interações dos agentes com 64
algoritmos de colônia de formigas 64
Figura 3.7: Quantidade máxima de ligações em grafos não-direcionados 65
Figura 3.8: Rede não direcionada com 8 estados 66
Figura 3.9: Sociomatriz do grafo da figura 3.8 66
Figura 3.10: Grafo para exemplificar a distância geodésica 68
Figura 4.1: Aprendizagem por reforço (Sutton e Barto, 1998) 75
Figura 4.2: Algoritmo *Q-learning* (adaptado de Watkins e Dayan (1992)) 81
Figura 4.3: Algoritmo *R-learning* (Schwartz, 1993) 83
Figura 4.4: Algoritmo *H-learning* (Tadepalli e Ok, 1994) 84
Figura 4.5: Comportamento de formigas reais (Goss *et al.* 1989) 91

Figura 5.1: Interação com informações partilhadas 107
Figura 5.2: Algoritmo de aprendizagem por reforço social 109
Figura 5.3: Modelos de compartilhamento de recompensas 110
Figura 5.4: Atualiza política 110
Figura 5.5: Diagrama de atividade do processo de aprendizagem 111
Figura 5.6: Exemplo de um ambiente com 400 estados. Os agentes são posicionados aleatoriamente no ambiente e possuem campo de profundidade visual de 1 113
Figura 5.7. Ambientes usados nas simulações 114
Figura 5.8: Modelo discreto 114
Figura 5.9: Modelo contínuo 115
Figura 5.10: Modelo dirigido por objetivo 116
Figura 5.11: Ambiente 400 estados, 3 agentes 117
Figura 5.12: Ambiente 400 estados, 5 agentes 117
Figura 5.13: Ambiente 400 estados, 10 agentes 117
Figura 5.14: Ambiente de 100 estados; 3 agentes 118
Figura 5.15: Ambiente de 100 estados; 5 agentes 118
Figura 5.16: Ambiente de 100 estados; 10 agentes 119
Figura 5.17: Ambiente de 250 estados; 3 agentes 119
Figura 5.18: Ambiente de 250 estados; 5 agentes 119
Figura 5.19: Ambiente de 250 estados; 10 agentes 120
Figura 5.20: Ambiente de 400 estados; 3 agentes 120
Figura 5.21: Ambiente de 400 estados; 5 agentes 120
Figura 5.22: Ambiente de 400 estados; 10 agentes 121
Figura 5.23: Pseudocódigo do *Ant-Q* (baseado em Gambardella e Dorigo, 1995) 123
Figura 5.24: Cálculo para AQ_0 124
Figura 5.25: Função *exploitation* 124
Figura 5.26: Função *exploration* 125
Figura 5.27: Atualização local 126
Figura 5.28: Atualização global 126
Figura 5.29: Ambientes usados na simulação, onde os estados estão expressos em um sistema euclidiano de coordenadas 2D 127
Figura 5.30: Evolução da política a cada 50 episódios 128
Figura 5.31: Eficiência da taxa de aprendizagem 129
Figura 5.32: Eficiência do fator de desconto 129

Figura 5.33: Resultados do parâmetro de exploração 130
Figura 5.34: Resultados da regra de transição (δ e β) 130
Figura 5.35: Quantidade de agentes (m_k) 131
Figura 5.36: Dinâmica do ambiente 133
Figura 5.37: Pseudocódigo do *Ant-Q* com as estratégias (modificado de 5.26) 136
Figura 5.38: Campo limite de 1; 10% de alterações a cada 100 episódios 138
Figura 5.39: Campo limite de 1; 20% de alterações a cada 100 episódios 138
Figura 5.40: Campo limite de 2; 10% de alterações a cada 100 episódios 138
Figura 5.41: Campo limite de 2; 20% de alterações a cada 100 episódios 139
Figura 5.42: Campo limite de 5; 10% de alterações a cada 100 episódios 139
Figura 5.43: Campo limite de 5; 20% de alterações a cada 100 episódios 139
Figura 5.44: Exemplo de políticas de ação 144
Figura 5.45: Eficiência da política em relação ao grau dos estados 145
Figura 5.46: Processo de crescimento da rede de relacionamentos 146
Figura 5.47: Grafos de relações, distâncias e feromônios 149
Figura 5.48: Políticas usadas para simular o crescimento da rede de relacionamento 149
Figura 5.49: Rede de relacionamentos em t_3 153
Figura 5.50: Diagrama de atividades 154
Figura 5.51: Distribuição dos estados no plano 156
Figura 5.52: Variações do custo das políticas com o *Ant-Q* e o *SAnt-Q* 158
Figura 5.53: *Ant-Q* vs. *SAnt-Q*, eil51 com 500 episódios 160
Figura 5.54: *Ant-Q* vs. *SAnt-Q*, eil76 com 500 episódios 161
Figura 5.55: Oscilação das políticas com o *SAnt* com q_0=1 após os episódios t_{30}, t_{50} e t_{100} .. 162
Figura 5.56: Soluções com o *Ant-Q* e *SAnt-Q* (eil51) 164
Figura 5.57: Soluções com o *Ant-Q* e *SAnt-Q* (eil76) 165
Figura 5.58: Políticas obtidas com o gerador de teste (eil51) 168
Figura 5.59: Políticas obtidas com o gerador de teste (eil76) 168
Figura 5.60: Evolução da rede com o método de otimização social (eil51) 170
Figura 5.61: Políticas obtidas com o método de otimização social (eil51) em 10.000 episódios 170
Figura 5.62: Evolução da rede com o método de otimização social (eil76) em 10.000 episódios 171
Figura 5.63: Políticas obtidas com o método de otimização social (eil76) em 10.000 episódios 172

Figura 5.64: *Ant-Q* sem heurística *vs.* método social, eil51 com 10000 episódios 173
Figura 5.65: *Ant-Q* sem heurística *vs.* Método Social, eil76 com 10000 episódios 173

Lista de Tabelas e Quadros

Tabela 2.1: Situações de interações 33
Tabela 5.1: Superioridade média do modelo híbrido em relação aos modelos 121
discreto, contínuo e dirigido por objetivo 121
Tabela 5.2: Estados antes e após as alterações 134
Tabela 5.3: Relações, distâncias e feromônios 149
Tabela 5.4: Influência de v_1 nas relações de $Q(r)$ em t_2 150
Tabela 5.5: Intensidade das $Q(r)$ em t_2 151
Tabela 5.6: Influência de v_2 nas relações de $Q(r)$ em t_3 151
Tabela 5.7: Intensidade das $Q(r)$ em t_3 152
Tabela 5.8: Valores da $Q(r)$ em t_1, t_2 e t_3 152
Tabela 5.9: Custo médio das melhores políticas (eil51) com 5000 episódios 159
Tabela 5.10: Custo médio das melhores políticas (eil76) com 5000 episódios 159
Tabela 5.11: Custo das políticas (eil51) 160
Tabela 5.12: Custo da política (eil76) 161
Tabela 5.13: *p-valor* com o teste de Friedman 166
Tabela 5.14: Comparativo das médias com o teste de Friedman (500 episódios) 166
Tabela 5.15: Custo médio das políticas do *Ant-Q* sem heurística e do método social (eil51 e eil76) 172 172

Lista de Abreviaturas

ADOPT	*Asynchronous Distributed Constraint Optimization*
ACS	*Ant Colony System*
DPOP	*Dynamic Programming OPtimization*
FANT	*Fast Ant System*
FSS	*Fish School Search*
GRADAP	*Graph Definition and Analysis Package*
MH	Método Híbrido
NCBB	*No-Commitment Branch and Bound*
NP	*Non-Deterministic Polynomial*
OptAPO	*Optimal Asynchronous Partial Overlay*
STEAM	*Simply, A Shell for Teamwork*
TAEMS	*Task Analysis, Environment Modeling and Simulation*
WWW	*World Wide Web*

Capítulo 1

Introdução

O comportamento coletivo de grupos sociais inspirou o desenvolvimento de modelos computacionais como geradores de soluções para problemas de otimização. Esse comportamento é resultado de padrões de interações entre os indivíduos da população, não sendo apenas propriedade de um único sistema de controle, mas deduzido de simples comportamentos individuais. Neste tipo de sistema, a estrutura de cada indivíduo é relativamente simples, mas a partir de seus comportamentos coletivos emergem estruturas sociais complexas.

Uma estrutura social é construída a partir de comportamentos e interações sociais, sendo que mudanças nesta estrutura produzirão efeitos em todos os indivíduos, eliminando comportamentos idiossincráticos. A interação social é o principal evento na construção de uma estrutura social, que mostra como os indivíduos estão relacionados. A estrutura social reflete as características e os comportamentos dos indivíduos, que podem influenciar ou alterar a estrutura social. A interação, por sua vez, é necessária porque na maioria dos sistemas sociais existem processos de adaptação dos indivíduos, que precisam se adaptar e favorecer o aperfeiçoamento de comportamentos coletivos, *i.e.*, do grupo.

Sistemas sociais são principalmente caracterizados pelas interações dos indivíduos de uma população, que interagem para formar comportamentos que melhorem a coordenação do grupo. Geralmente, os indivíduos aprendem a partir do seu próprio comportamento individual e das influências dos demais indivíduos, no intuito de melhorar a sua utilidade a partir da interação com os indivíduos mais fortes do grupo. A estrutura social pode ser determinada pela sobreposição dos melhores indivíduos do sistema, onde indivíduos com maior força influenciam os demais através de recompensas individuais ou coletivas, seguindo os princípios da teoria do impacto social (Latané, 1981) e da aprendizagem por reforço.

A partir do comportamento dos indivíduos e da aplicação dessas teorias é possível identificar estruturas sociais coerentes, padrões e comportamentos de um sistema complexo, descrevendo quem interage com quem, a frequência e a intensidade de interação entre eles. A estrutura social emerge sem um sistema central de coordenação, mas da sociabilidade dos indivíduos a partir de comportamentos autônomos e locais.

A estrutura social com essas teorias pode formar um sistema multiagente, onde as interações entre os indivíduos são utilizadas para que os mesmos alcancem seus objetivos individuais e coletivos. O desempenho dos indivíduos pode então depender fortemente da estrutura do sistema social e de questões relacionadas ao impacto das relações na atuação de cada indivíduo e à socialização das recompensas. O impacto das relações pode ser observado a partir de técnicas de análise das redes sociais como um modelo para a avaliação de uma estrutura social, construída à medida que as interações dos indivíduos ocorrem com as recompensas geradas.

Uma característica importante dos indivíduos sociais que formam num único sistema é a capacidade da coordenação enquanto interagem com os demais indivíduos. Essa é uma característica importante quando vários indivíduos estão inseridos em um ambiente compartilhado. Em um sistema multiagente, os indivíduos precisam interagir e se coordenar para a execução das tarefas. A coordenação entre indivíduos pode ajudar a evitar problemas como soluções redundantes, inconsistência de execução, desperdício de recursos e espera por eventos que provavelmente não irão ocorrer. Neste contexto, modelos de coordenação baseados em enxames têm se mostrado adequados para soluções de problemas complexos integrando comportamentos sociais e individuais.

O paradigma de coordenação baseado em inteligência de enxames tem sido intensamente estudado nessa última década (Kennedy e Eberhart, 2001). Esse paradigma é inspirado nas colônias de insetos sociais, onde sistemas computacionais reproduzem os comportamentos utilizados para a resolução de problemas coletivos em colônias de formigas, abelhas, cupins ou vespas.

Os insetos sociais de um enxame atuam localmente, mas devem satisfazer o objetivo global do sistema (comunidade de agentes). A comunidade pode ser formada por um conjunto de indivíduos e as conectividades indicam suas relações sociais. Essa descrição também é apontada como o conceito fundamental de redes sociais (Wasserman e Faust, 1994). As redes sociais podem representar um conjunto de indivíduos, computadores, organizações ou elementos computacionais que estão conectados por algum tipo de relação. Por exemplo, um conjunto de pessoas pode estar ligado por relações de amizade, de conhecimento, de parentesco ou

de trabalho, assim como insetos de um enxame podem estar relacionados devido à sua proximidade geográfica, tipo de especialização ou tarefa comum.

A coordenação destes indivíduos pode ser melhorada quando conceitos de redes sociais são utilizados para direcionar o compartilhamento de informações, aprimorando os algoritmos baseados em enxames ou aprendizagem por reforço, intensificando relações para melhorar o comportamento individual e coletivo. Esses comportamentos são mantidos por valores que determinam as atitudes dos indivíduos, sendo denominados de **recompensas sociais por interações** e capazes de influenciar ou alterar a estrutura social, modificando a coordenação dos indivíduos.

Neste trabalho defendemos a ideia de que essas recompensas podem ser socializadas com modelos específicos de compartilhamento de recompensas e princípios de redes sociais. Um método que integra aprendizagem por reforço, sistemas multiagente, teorias sociais e otimização foi desenvolvido para atuar em estruturas sociais dinâmicas. O método proposto foi denominado de *SAnt-Q* (*Social Ant-Q*) e é uma das principais contribuições desta tese.

1.1 Problema

Coordenar indivíduos com comportamentos diferentes é um importante tema de estudo em sistemas multiagente. Técnicas de coordenação derivadas da aprendizagem por reforço vêm sendo estudadas por diversos pesquisadores e descritas em diferentes aplicações (Kaelbling *et al.* 1996; Sutton e Barto, 1998). A aprendizagem por reforço ocorre quando um indivíduo aprende por tentativa e erro ao interagir com o ambiente e com outros indivíduos. A fonte de aprendizado é a própria experiência do indivíduo, cujo objetivo é adquirir comportamentos que melhorem a estrutura social através das recompensas adquiridas nas interações.

Técnicas de análise das redes sociais podem identificar relações escondidas entre os indivíduos que nelas interagem. Para analisar o impacto das relações dos indivíduos e das recompensas sociais compartilhadas, é possível utilizar a teoria dos grafos que permite a visualização do ambiente e análises numéricas. As interações observadas geralmente apresentam uma forte influência nas relações dos indivíduos e vice-versa. Essas relações também podem ser criadas, reforçadas ou enfraquecidas com técnicas de aprendizagem por reforço e algoritmos baseados em enxames apoiados em alguma teoria de análise social.

É possível observar que cada um dos indivíduos em um sistema multiagente sofrem influências dos demais indivíduos, mas até o presente momento, pouca pesquisa tem sido realizada sobre a formalização desse processo e a construção de estruturas sociais dinâmicas de

tomada de decisão com o objetivo de aprimorar os métodos de coordenação e aprendizagem distribuídas existentes na literatura. Portanto, os métodos desenvolvidos neste trabalho devem responder as seguintes perguntas:

- Como **formalizar** essas **influências**?
- Como **identificar** os **indivíduos relevantes**?
- Como acrescentar a **dimensão social** aos **modelos de recompensas** existentes?

Outra questão importante a ser considerada consiste na utilização dos princípios das redes sociais para melhorar a coordenação dos indivíduos que compartilham recompensas sociais. A partir dessas observações, novos questionamentos podem ser formulados:

- Como **construir** uma **rede de relacionamentos** a partir do conhecimento adquirido pelos indivíduos ao longo das interações? e,
- Como **utilizar** os princípios sociais para gerar modelos de compartilhamento de **recompensas dos indivíduos**?

As pesquisas apresentadas neste trabalho vão ao encontro dessas questões apresentando metodologias desenvolvidas para esse fim.

1.2 Hipóteses

As questões levantadas anteriormente podem ser estudadas pragmaticamente a partir da adaptação de métodos de coordenação multiagente, colônia de formigas, aprendizagem por reforço e redes sociais.

Algoritmos baseados em população inspirada no comportamento das colônias de formigas constituem uma forma coletiva de coordenação entre indivíduos. Por outro lado, indivíduos com algoritmos de aprendizagem por reforço devem estabelecer, de maneira autônoma e interativa, políticas de ação e/ou comportamentos (mapeamento estado-ação), mapeando o espaço de estados e controlando o comportamento global do sistema. Algoritmos de aprendizagem por reforço têm inspirado nestes últimos anos o desenvolvimento de algoritmos de colônia de formigas que recebem recompensas quando objetivos pré-estabelecidos são alcançados. As recompensas acabam reforçando as relações existentes entre os estados do sistema.

Por outro lado, é possível observar o impacto das relações estabelecidas através da aplicação da teoria das redes sociais. Portanto, acreditamos que métodos de inteligência de enxames, aprendizagem por reforço e os modelos de sistemas sociais estão baseados em princípios muitas vezes complementares, possibilitando a adaptação e desenvolvimento de métodos

de coordenação para auxiliar na resolução de problemas de larga escala (muitos indivíduos) que exigem distribuição e coordenação das ações.

Neste processo, é fundamental analisar as redes de relacionamentos construídas ao longo do processo de interação para melhorar a qualidade e aumentar a eficiência da coordenação. Além disso, em sistemas multiagente com algoritmos de colônia de formigas e aprendizagem por reforço, um indivíduo pode ser influenciado pelas recompensas geradas por outros indivíduos, sendo necessário o desenvolvimento de metodologias de tomada de decisão, apoiadas na teoria das relações sociais, que alteram comportamentos individuais e as relações estabelecidas durante as interações.

Acreditamos que com os conceitos da teoria e da análise das redes sociais e a estrutura social construída a partir das interações, é possível melhorar a coordenação dos indivíduos de um sistema, sendo que tal metodologia de coordenação poderia reduzir o tempo necessário para a convergência de modelos baseados em recompensas, além de reduzir problemas de escalabilidade, favorecendo a resolução de problemas de otimização combinatória.

1.3 Objetivos

Este trabalho possui dois objetivos principais:

(i) Conceber modelos para compartilhamento de recompensas sociais; e

(ii) Utilizar a estrutura social construída com a sociabilidade dos indivíduos para melhorar o comportamento social de um sistema multiagente;

Para atingir esses objetivos, vislumbramos ainda a realização dos seguintes objetivos específicos:

- Estudar o impacto das recompensas sociais por interações em problemas de coordenação multiagente;
- Estudar o impacto das redes sociais em problemas de otimização por colônia de formigas;
- Estudar o impacto das recompensas compartilhadas pelos indivíduos; e
- Avaliar os modelos concebidos em problemas de otimização combinatória.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: No capítulo 2 são apresentados e comparados os principais métodos de coordenação. No capítulo 3 são apresentados conceitos

básicos das redes sociais e fundamentos dos grafos. Esse capítulo é finalizado com uma discussão sobre a relação existente entre a teoria das redes sociais e os sistemas multiagente. Já no capítulo 4 são discutidos os principais conceitos sobre aprendizagem por reforço e otimização por colônia de formigas, bem como seus principais algoritmos. O capítulo 5, por sua vez, apresenta o enfoque proposto (metodologia) e as etapas de desenvolvimento dos métodos propostos, ilustrando os algoritmos e discutindo os resultados experimentais. Na sequência são apresentadas as conclusões e discussões finais do trabalho.

Capítulo 2

Aprendizagem e Coordenação em Sistemas Multiagente

A aprendizagem e coordenação de agentes vêm recebendo grande atenção por parte da comunidade da inteligência artificial. Mesmo em aplicações aparentemente simples torna-se muitas vezes difícil ou mesmo impossível prever comportamentos que garantam a um agente um desempenho aceitável ao longo de todo o seu ciclo de vida. Em razão desta dificuldade, é geralmente necessário adaptar agentes com alguma capacidade de aprendizagem que lhes permitam modificar seu comportamento em função da experiência adquirida e do possível modelo de coordenação disponível. A coordenação, neste caso, é necessária para garantir comportamentos globalmente coerentes para sistemas formados por indivíduos que compartilham objetivos, recursos e habilidades. Este capítulo apresenta uma introdução sobre agentes, aprendizagem e coordenação em sistemas multiagente. Agentes são utilizados neste trabalho para simular e avaliar os métodos de coordenação existentes. Ao longo desde capítulo, também são apresentados e comparados alguns dos principais métodos de coordenação multiagente.

2.1 Agentes Inteligentes

A área de agentes é integrada por pesquisadores de diferentes áreas como, inteligência artificial, sistemas distribuídos, interface homem-computador e robótica, que juntos têm como principais objetivos: atender aos novos requisitos exigidos por determinadas aplicações; facilitar a interação usuário/máquina e; construir sistemas inteligentes. A partir das definições observadas nos trabalhos de alguns pesquisadores é possível perceber que vários deles possuem diferentes opiniões para o termo agente (Maes, 1995; Wooldridge e Jenning, 1995; Castel-

franchi, 1996). Dessa forma, não existe uma definição unânime dentre os pesquisadores sobre o conceito “agente”, porém, muitas destas definições estudadas se complementam.

Maes (1995) e Hendler (1996) consideram agentes inteligentes os programas de inteligência artificial cuja finalidade é agir em diversos ambientes de importância para os seres humanos, sendo divididos em duas categorias: agentes físicos e agentes de informação. Os agentes físicos trabalham em um ambiente onde é difícil inserir um ser humano (*e.g.*, espaço) ou que seja perigoso (*e.g.*, núcleo de um reator nuclear). Os agentes de informação atuam em um mundo virtual onde existe uma grande quantidade de informações espalhadas por diversos computadores (*e.g.*, Internet).

Wooldridge (1999) divide as aplicações de agentes em dois grupos. O primeiro, sob uma notação mais fraca, na qual agentes compõem o hardware ou, geralmente, softwares dotados de *autonomia* para a realização de suas tarefas, *habilidades* para interagir com outros agentes e entidades, e *reatividade* ao meio em que está inserido, a qual geralmente aumenta o grau de dinamismo e complexidade. Sob um contexto mais complexo, o segundo agrupamento se baseia em uma notação mais forte (utilizada por pesquisadores da inteligência artificial), a qual define um agente como um software que, além das propriedades anteriores, implementam conceitos geralmente aplicados aos seres humanos, tais como o conhecimento, a crença, a intenção e a obrigação.

No geral, um agente é definido como uma entidade de software que exibe comportamentos autônomos e está situado em algum ambiente sobre o qual é capaz de realizar ações para alcançar seu próprio objetivo. O termo ambiente refere-se a uma representação do sistema estudado, onde os agentes são simulados. A figura 2.1 mostra uma representação abstrata entre os agentes, na qual a percepção e a interação podem possibilitar as ações e a troca de informações.

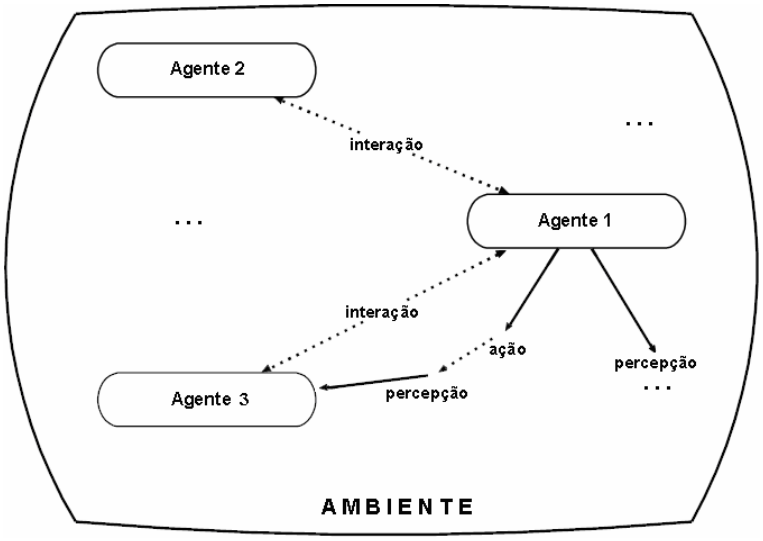


Figura 2.1: Modelo abstrato de agentes inseridos em um ambiente

Wooldridge (1999) descreve algumas características típicas de agentes inteligentes:

i) Reação: agentes devem perceber seu ambiente e responder oportunamente às mudanças que nele ocorrem;

ii) Pró-atividade: agentes não devem simplesmente atuar em resposta ao ambiente, devem exibir um comportamento oportunista e direcionado ao seu objetivo e tomar a iniciativa quando apropriado;

iii) Sociabilidade: agentes devem interagir, quando apropriado, com outros agentes artificiais ou humanos para completar suas próprias soluções de problemas ou ajudar outros com suas atividades.

Bradshaw (1997) acrescenta ainda outras propriedades que os agentes devem possuir para se diferenciar de simples programas de computadores:

i) Capacidade de inferência: agentes podem agir sobre especificações abstratas de tarefas utilizando um conhecimento anterior, conseguindo ir além das informações fornecidas, e devem possuir algum modelo de si próprio, de usuários, de situações de outros agentes;

ii) Continuidade: agentes que conseguem fazer persistir a sua identidade e estados durante longos períodos de tempo;

iii) Adaptabilidade: agentes são capazes de aprender e melhorar com a experiência; e

iv) Mobilidade: agentes podem migrar de forma intencional de um determinado local para outro.

Outro importante aspecto é a autonomia dos agentes. Um agente inteligente interage de forma autônoma quando tem a capacidade de executar o controle sobre suas próprias ações em seu ambiente de interação. Os agentes que conseguem melhorar seu comportamento em função de suas ações anteriores são definidos como agentes autônomos adaptativos (Enembreck, 2003). Um dos desafios da inteligência artificial consiste em criar sistemas capazes de melhorar seu desempenho a partir de suas experiências. Esta capacidade de adaptação é fundamental para os sistemas cujo comportamento é autônomo, pois também pode promover economia de recursos e aumentar a confiabilidade.

No início dos anos 90, certo cepticismo reinava sobre a utilização de mecanismos de aprendizagem em sistemas dinâmicos como aqueles dos agentes autônomos, devido à incipiência das pesquisas efetuadas no domínio da aprendizagem automática. Nesta época, algoritmos de aprendizagem necessitavam de recursos computacionais até então raros além de uma quantidade enorme de dados, como as redes neurais, por exemplo. Felizmente, as técnicas de aprendizagem e as tecnologias computacionais foram aprimoradas. No entanto, nem todos os algoritmos de aprendizagem são indicados para um agente autônomo adaptativo, porque esses devem ter as seguintes características (Enembreck, 2003):

i) a aprendizagem deve ser incremental;

ii) deve levar em conta o ruído;

iii) a aprendizagem não poderá ser supervisionada;

iv) eventualmente é necessário que o algoritmo permita a utilização de conhecimentos fornecidos pelo usuário e/ou por quem o desenvolveu.

Enembreck (2003) completa ainda, que certas formas de adaptação podem ser vistas como aprendizado a partir da experiência. Neste caso, o agente irá melhorar à medida que o tempo passa. Isso significa que o agente deve aprender a escolher as boas ações nos bons momentos, com uma melhoria constante no mecanismo de seleção de ações.

2.2 Coordenação dos Agentes

Em sistemas multiagente a coordenação dos agentes é necessária para o aumento da qualidade de soluções produzidas, melhorando o processo durante a resolução das tarefas realizadas pelos agentes. Os benefícios da resolução distribuída de problemas são anulados quando a coordenação é deficiente, podendo causar interação desordenada entre os agentes. Dessa

forma, agentes sem coordenação podem agir sem coerência e entrar em conflito com seus próprios recursos inviabilizando a convergência entre objetivos locais e globais (Jennings e Bussmann, 2003). Os principais requisitos para a coordenação são citados em (Durfee, 1988): i) comunicação entre os agentes; ii) reconhecimento das interações potenciais dos planos; e iii) negociação entre os agentes.

Quando agentes autônomos atuam no mesmo ambiente é necessário gerenciar as tarefas complementares, que permitem a troca de informações para ocorrer o processo de coordenação (Ferber, 1999). O gerenciamento é necessário porque muitas vezes agentes precisam de informações e resultados disponíveis em outros agentes, necessitando da coordenação para que haja a troca do conhecimento entre eles, possibilitando que problemas como uso ineficiente de recursos e atividades desnecessárias e redundantes sejam evitadas.

A coordenação é desejável devido a diferentes fatores, como (i) a dependência das ações dos agentes, pois um único agente geralmente não possui a competência, e (ii) a distribuição dos recursos ou das informações necessárias para resolver problemas complexos de forma independente. Além disso, agentes podem ter objetivos e ações antagônicos, que podem contribuir para o fracasso da interação, existência de restrições globais à solução de problemas e existência de procedimentos que satisfaçam os objetivos individuais e globais quando ações ou tarefas são executadas de maneira conjunta.

Em outras situações, algumas ações quando executadas simultaneamente podem levar a conflitos, ou produzir efeitos positivos como a melhora no desempenho global do sistema. Ferber (1999) cita que as relações podem ser negativas ou positivas (figura 2.2). As relações negativas ou relacionamentos conflitantes impossibilitam a execução de algumas ações, que podem ser causadas por incompatibilidade de objetivos, ou pela limitação dos recursos disponíveis. Já nas relações positivas ou relacionamentos cooperativos, as ações se favorecem mutuamente, resultando em maior eficiência do que se fossem executadas independentemente.

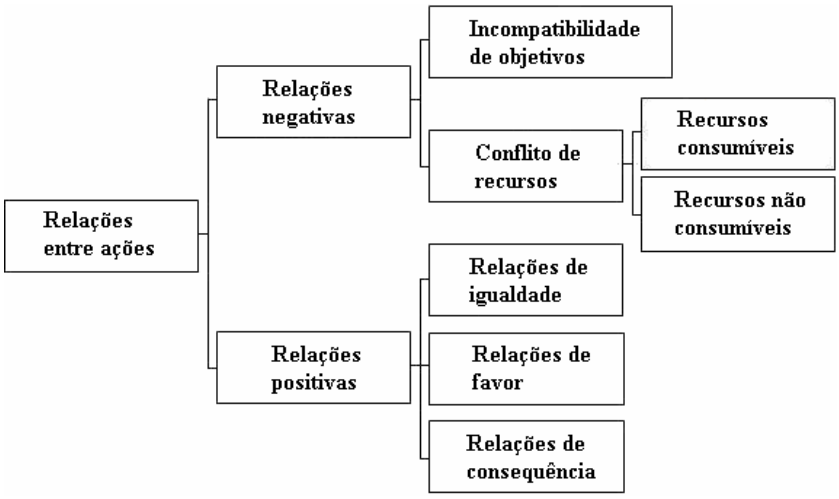


Figura 2.2: Tipos de relações entre ações (Ferber, 1999)

Moulin e Chaib-Draa (1996) descrevem três processos fundamentais para a coordenação. Primeiro, o *ajuste mútuo*, uma maneira de coordenação que pressupõe que dois ou mais agentes concordam em compartilhar recursos para atingir um objetivo. Segundo, a *supervisão*, na qual existem relações estabelecidas entre os agentes, na qual um agente mantém algum controle sobre os outros. Terceiro, a *padronização*, que estabelece uma relação entre os agentes, na qual um agente mantém o controle sobre os demais (agentes coordenados) estabelecendo procedimentos padronizados que serão seguidos pelos coordenados em determinadas situações.

Esses processos podem ser definidos como relações de dependência em relação aos outros agentes (Castelfranchi *et al.* 1992; Sichman, 2003). Por exemplo, um agente ag_i é dito autônomo para o objetivo g_m se e somente se: (i) ag_i deseja atingir o objetivo g_m; (ii) existe um plano p_s cuja execução atinja g_m tal que todas as suas ações podem ser desempenhadas por ag_i. Caso não seja autônomo para um dado objetivo, um agente ag_i é dito dependente para este objetivo. O fato de ser dependente, porém, não significa que exista necessariamente um agente que possa executar a ação de que ag_i necessita. Esta situação é representada por uma *relação de dependência*. Assim, um agente ag_i é dito dependente de um agente ag_j para o objetivo g_m se e somente se: (i) ag_i tem o objetivo g_m; (ii) ag_i é dependente para o objetivo g_m; (iii) existe um plano p_s cuja execução atinja g_m e no qual ag_j pode realizar alguma ação de que ag_i necessita.

Os modelos de racionalidade também favorecem o processo de coordenação, auxiliando na determinação de quais ações realizar primeiro, quais objetivos a serem atingidos e com quem se relacionar. Um dos modelos mais comuns é baseado na utilidade, que toma como

princípio da racionalidade o utilitarismo. Nesse paradigma, um agente é dito racional caso sempre busque a maximização de sua utilidade esperada (Doyle, 1992 *apud* Sichman, 2003). Essa noção de racionalidade encontra-se presente na maior parte das teorias econômicas modernas. Outro modelo é baseado na complementaridade. Nesse modelo as escolhas dos agentes para interagir com outros são fundamentadas em relações estruturais objetivas nas quais os agentes encontram-se inseridos. Uma dessas relações fundamentais é a relação de dependência social (Castelfranchi *et al*, 1992). Nessa relação, os agentes quase sempre necessitam uns dos outros para atingirem seus objetivos, e quando estas relações de dependências tornam-se subjetivas, eles podem explicar por que agentes adotam os objetivos uns dos outros e por que algumas interações sociais surgem do seio de uma sociedade.

O conceito de coordenação e racionalidade define aspectos gerais de interação dos agentes, de maneira a viabilizar ações em relação ao objetivo global. A figura 2.3 apresenta uma classificação para o processo de coordenação.

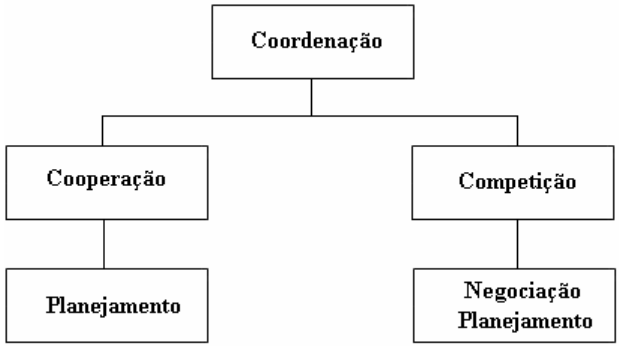


Figura 2.3: Classificação da coordenação (Moulin e Chaib-Draa, 1996)

Ao desenvolver um sistema multiagente, é desejável que os elementos que formam a classificação da coordenação sejam considerados, na intenção de compatibilizar as ações dos agentes. Os elementos que compõem a classificação da coordenação da figura 2.3 são apresentados a seguir.

a) Cooperação

A cooperação ocorre quando *n* agentes planejam e executam suas ações de maneira coordenada, na intenção de solucionar problemas para os quais tenham sido modelados. A cooperação é desejável quando:

i) O agente não encontra um plano local que satisfaça os objetivos;

ii) O plano disponível envolve ações de outros agentes; e

iii) O agente considera que um plano externo pode ser melhor (menor custo ou mais eficiente) do que um plano local;

Durante a fase de planejamento pode-se encontrar ainda outras situações:

i) O agente encontra planos incompletos, que podem ser completados em cooperação com outros agentes; ou

ii) Quando o agente enfrenta situações para o qual não esteja capacitado, mas entende que outros agentes podem ser capazes de tratá-las.

A cooperação entre os agentes oferece as seguintes vantagens (Moulin e Chaib-Draa, 1996):

i) otimização do tempo de execução de uma tarefa;

ii) aumento do escopo de tarefas executáveis através do compartilhamento de recursos;

iii) maior probabilidade de finalização de uma tarefa; e

iv) diminuição da interferência entre as tarefas, evitando interações desnecessárias.

b) Planejamento

O processo de planejamento constitui uma forma especializada de processo de cooperação, produzindo um conjunto de atividades organizadas com um curso de ação definido, na qual estas atividades são distribuídas aos agentes capacitados a executá-las. O planejamento pode acontecer de maneira centralizada ou distribuída. Centralizada quando um único agente é responsável em desenvolver um plano e distribuída quando pressupõe que o plano seja desenvolvido por mais de um agente, sendo considerada quando um único agente não possui uma visão global das atividades do grupo.

Segundo Durfee (1996) o planejamento em sistemas multiagente consiste em três etapas:

i) formulação de um curso de ação, considerando as ações a serem executadas em paralelo pelos demais agentes;

ii) identificação do curso de ação de outros agentes; e

iii) identificação da maneira pela qual um agente poderia comprometer-se com seus próprios modelos.

Pode ocorrer que os agentes necessitem ajustar seus planos, devido a motivos como: i) resultados de suas próprias ações; ii) resultados de ações de outros agentes; iii) alterações no

ambiente; iv) alterações de objetivos; e v) alterações na percepção do agente quanto ao contexto multiagente no qual está inserido.

Técnicas como planejamento centralizado, reconciliação de planos, planejamento distribuído e análise organizacional são alternativas para auxiliar as atividades dos agentes em determinar tarefas, após raciocinar sobre as consequências destas em certas organizações.

c) Negociação

A negociação é importante nas atividades cooperativas dentro das sociedades humanas, pois permite que pessoas resolvam conflitos que possam interferir no comportamento cooperativo (Moulin e Chaib-Draa, 1996). Segundo Huhns e Stephens (1999), os principais elementos utilizados pelos agentes envolvidos no processo de negociação são:

i) Linguagem;

ii) Protocolo que define a maneira que os agentes negociam; e

iii) Processo de decisão que determina suas posições, concessões e critérios utilizados para os acordos.

A negociação pode ter as seguintes abordagens:

i) Centradas no ambiente: o mecanismo de negociação deve possuir suas próprias regras, na intenção de interagir com os demais agentes de maneira produtiva e razoável. Nesse caso, as principais propriedades são:

- Eficiência: os agentes devem otimizar recursos para alcançar determinados acordos;
- Estabilidade: todos os agentes devem cumprir os acordos;
- Simplicidade: baixas demandas computacionais e de comunicação devem ser impostas aos agentes;
- Distribuição: realizar decisões de maneira descentralizada; e
- Simetria: não deve haver diferenciação (benefícios) entre os agentes.

ii) Centradas nos agentes: assume que os agentes sejam racionais e o conjunto deles é reduzido, pois necessitam de uma linguagem e abstração do problema comum. Podem ser utilizados protocolos de negociação unificados onde agentes criam um acordo que constitui um plano conjunto para satisfazer os objetivos.

Do processo de negociação, algumas situações podem surgir (Rosenschein e Zlotkin, 1994), por exemplo: (i) conflito, quando o conjunto de negociações é nulo; (ii) compromisso,

quando os agentes preferem trabalhar de maneira isolada, caso contrário, tentam chegar ao acordo negociado; e (iii) cooperativo, quando todos os acordos do conjunto de negociação são desejados pelos agentes.

Uma abordagem bastante utilizada em sistemas multiagente é o protocolo de redes de contrato proposto por (Smith, 1980), inspirado nos processos de contratação existentes em organizações humanas. Neste processo, agentes coordenam suas ações através de contratos para cumprir seus objetivos específicos, onde existe um agente que atua como gerente, decompondo seus contratos em subcontratos a serem realizados por outros agentes potenciais executores. Da perspectiva do gerente, o processo consiste em (Huhns e Stephens, 1999):

i) Anunciar uma tarefa que precisa ser executada;

ii) Receber e avaliar ofertas dos agentes executores potenciais;

iii) Alocar um contrato para um executor apropriado; e

iv) Receber e sintetizar os resultados.

A partir da perspectiva do executor, o processo é: (i) receber anúncios de tarefa; (ii) avaliar a própria capacidade de resposta; (iii) responder (recusa, oferta); (iv) executar a tarefa se a oferta enviada foi aceita; e (v) enviar resultados ao gerente.

O protocolo de redes de contrato oferece a vantagem de degradação suave do desempenho. Se um executor não está apto a prover uma solução considerada satisfatória, o gerente pode procurar outros agentes executores potenciais para a tarefa. Outros modelos de negociação baseados em mercados econômicos (Raiffa, 1985) podem ser encontrados em (Faratin, 1998) e estratégias de negociação para sistemas multiagente são descritas em (Kraus, 2001).

Os primeiros trabalhos sobre negociação entre agentes foram propostos por Rosenschein e Genesereth em 1985, e Sycara em 1988 e 1990. O sistema denominado *persuader* (Sykara, 1990) foi implementado para operar no domínio da negociação das leis de trabalho. O sistema possuía três agentes, inspirados na negociação humana. O sistema permitia a troca interativa de propostas e contrapropostas para que os agentes chegassem a um acordo. A negociação envolvia várias questões, tais como salários, pensões, tempo de serviço, contratos de serviços, e assim por diante. A revisão de crenças para alterar a utilidades dos agentes era realizada por argumentação persuasiva. Além disso, técnicas de aprendizagem baseadas em casos também foram incorporadas ao modelo.

Diante do exposto nesta seção, vários métodos são apresentados para a coordenação e aprendizagem dos agentes. As subseções 2.3.1 à 2.3.7 discutem alguns dos principais métodos que apresentam esses princípios.

2.3 Métodos de Coordenação e Aprendizagem para Sistemas Multiagente

A aprendizagem em sistemas multiagente, diferentemente da aprendizagem em ambiente com um único agente, supõe que o conhecimento relevante não está disponível localmente em um único agente (Modi e Shen, 2001). Na aprendizagem multiagente, os agentes aprendem a realizar uma tarefa que envolve mais do que um agente na sua execução. Segundo Weiss e Sen (1996) a aprendizagem pode ser dividida de duas formas: a aprendizagem isolada e a aprendizagem coletiva. Na aprendizagem isolada, o processo de aquisição do conhecimento pelo agente ocorre sem a influência dos demais agentes ou qualquer outro elemento da sua sociedade. Já na aprendizagem coletiva, o processo de aquisição do conhecimento tem influência direta de todos os elementos da sociedade em que o agente está inserido.

Stone e Veloso (1996) completam ainda que, se um agente está aprendendo a conquistar habilidades para interagir com outros agentes em seu ambiente, e independentemente se os outros agentes estão ou não aprendendo simultaneamente, esta aprendizagem é considerada aprendizagem multiagente. Dessa forma, aprendizagem multiagente inclui algumas situações na qual o agente aprende interagindo com outros agentes, alterando e evoluindo o próprio modelo de coordenação.

2.3.1 Coordenação por Interação

A interação propicia a combinação de esforços entre um conjunto de agentes na busca de soluções para problemas globais, pressupondo ações de coordenação entre os agentes (DeLoach e Valenzuela, 2007). Alguns aspectos podem ser considerados no processo de interação dos agentes:

i) Quais agentes devem interagir;

ii) Em que momento ocorrerá a interação;

iii) Qual o conteúdo da interação ou comunicação;

iv) Como será realizada a interação, definindo os processos e recursos a serem utilizados;

v) Definir se a interação é necessária; e

vi) Empregando algum mecanismo, de que maneira será estabelecida a compreensão mútua (linguagem comum, interpretação baseada no contexto, *etc.*).

Uma situação de interação é um conjunto de comportamento resultantes de um grupo de agentes que agem para satisfazer seus objetivos, e que levam em conta as restrições devidas à limitações de recursos e à limitação de suas competências individuais (Ferber, 1999).

Considerando que a interação entre agentes pode ocorrer através de ações para atingir seus objetivos, agentes realizam ações, que podem eventualmente utilizar recursos, consumíveis ou não, que se encontram disponíveis no ambiente. Tais situações de interação podem ser classificadas de acordo com as dimensões distintas (Sichman, 2003):

i) Compatibilidade de objetivos: os objetivos dos agentes são considerados compatíveis/incompatíveis quando o fato de atingir um deles não acarretar/acarretar necessariamente a impossibilidade de atingir o outro;

ii) Quantidade de recursos: os recursos são considerados suficientes/insuficientes quando os agentes puderem/não puderem realizar suas tarefas simultaneamente;

iii) Competência dos agentes: a competência de um agente é considerada suficiente/insuficiente quando ele for capaz/incapaz de realizar sua tarefa, de modo a tingir seu objetivo.

A tabela 2.1 apresenta as possíveis situações de interações segundo tais dimensões, conforme Ferber (1999).

Tabela 2.1: Situações de interações

Objetivos	Recursos	Competências	Situações de interações
Compatíveis	Suficientes	Suficientes	Independência
Compatíveis	Suficientes	Insuficientes	Colaboração simples
Compatíveis	Insuficientes	Suficientes	Obstrução
Compatíveis	Insuficientes	Insuficientes	Colaboração coordenada
Incompatíveis	Suficientes	Suficientes	Competição individual pura
Incompatíveis	Suficientes	Insuficientes	Competição coletiva pura
Incompatíveis	Insuficientes	Suficientes	Conflito individual por recursos
Incompatíveis	Insuficientes	Insuficientes	Conflito coletivo por recursos

Fonte: Ferber (1999)

Muitas vezes, a interação entre os agentes está diretamente relacionada a um mecanismo de aprendizagem. Em sistemas multiagente a aprendizagem está diretamente relacionada com a maneira que os agentes se interagem, podendo comprometer a convergência parcial ou total da aprendizagem dos agentes ou até mesmo causar situações inexplicáveis nas suas ações, devido a conflitos entre comportamentos e objetivos e à limitação de recursos e habilidades. Por exemplo, em cenários clássicos como monitoramento com sensores distribuídos

(Conway *et al.* 1983), alocação distribuída de tarefas (Rosenschein e Zlotkin, 1994), e formação de coalizão (Sandholm *et al.* 1998), cada agente percebe uma parte do estado global do cenário e toma medidas que modificam alguma parte deste estado, na intenção de maximizar uma função de utilidade local (Vidal, 2004).

Dessa maneira, os agentes devem ser capazes de interagir em um ambiente comum, trocando informações relevantes e cooperando com os indivíduos que podem contribuir para conquistar um determinado objetivo. Na literatura são encontrados diversos trabalhos que descrevem diferentes formas de aprendizagem a partir da interação (Ferber, 1999; Wooldridge, 2002), aprendizagem coletiva ou social (Sichman, 2003).

Um modelo de aprendizagem por interação possui um conjunto de comportamentos resultantes do grupo de agentes que agem para satisfazer seus objetivos e ainda consideram as restrições impostas pela limitação de recursos e pelas competências individuais (Ferber, 1999). Em problemas de aprendizagem usando métodos de aprendizagem por reforço, a interação depende de um modelo que possibilita a troca das melhores recompensas acumuladas e dos reforços imediatos da transição. Com esse objetivo, Chapelle *et al.* (2002) propuseram um modelo por interações onde o valor da recompensa é calculado usando a satisfação individual dos agentes vizinhos. No processo de aprendizagem os agentes continuadamente emitem um nível de satisfação pessoal. Por exemplo, se a ação do agente *A* no ambiente *E* pode ajudar o agente *B*, o nível da satisfação de *B* também aumenta. O processo de aprendizagem ocorre até que todos os agentes vizinhos alcancem um nível satisfatório para as recompensas recebidas.

Em um trabalho anterior (Ribeiro *et al.* 2006a) foi desenvolvida uma estratégia de aprendizagem na qual os agentes inseridos no sistema são capazes de manter na memória políticas aprendidas para serem reusadas nas políticas futuras, evitando atrasos ou falta de convergência na aprendizagem dos agentes com a dinâmica no ambiente. O método proposto, denominado política adaptativa baseada em recompensas passadas (*K-learning*), foi testado em cenários com características de trânsito com diferentes níveis de congestionamento, usando de 3 a 10 agentes em ambientes com até 100 estados. Resultados experimentais mostraram que o método proposto é melhor do que o algoritmo *Q-learning* padrão, pois estima valores e encontra soluções usando políticas passadas. O método também foi testado em ambientes dinâmicos e com espaço de estados de tamanho da ordem de centenas (Ribeiro *et al.* 2009b).

Mataric (1998) propôs um método onde os agentes podem, de forma cooperativa, transmitir para outros agentes a situação atual do estado alterado, após a realização de uma tarefa. Neste caso, um agente somente poderá dividir seu aprendizado com o agente mais pró-

ximo do seu estado atual, a fim de economizar recursos e evitar troca de informações incorretas.

DeLoach e Valenzuela (2007) propuseram um mecanismo chamado de modelo de capacidade. O modelo visava demonstrar como os agentes interagiam em um dado ambiente, colocando em evidência o uso de suas capacidades. Esse modelo é composto pelos seguintes elementos: um modelo de capacidade, um ambiente e um conjunto de interações entre os objetos do modelo de capacidade e do ambiente. Neste caso, cada agente tem a capacidade de perceber e manipular os objetos do ambiente por meio de interações, aprendendo/refinando estratégias para alcançar seu objetivo. O modelo de capacidade define as ações possíveis que cada agente pode realizar com o objetivo de manipular os objetos do ambiente. Quando executada uma ação, essa atividade recebe uma recompensa do ambiente. Se a ação modifica o objeto, o ambiente consequentemente é alterado com as recompensas recebidas.

Na maioria das vezes é difícil adaptar os métodos propostos em um modelo genérico de coordenação, devido à diversidade das classes de problemas existentes e o demasiado conhecimento do domínio exigido. Além disso, essas interações podem não compartilhar as melhores informações de algoritmos baseados em recompensas, pois não consideram a reputação de cada agente e acabam ocasionando a troca de informações não satisfatórias.

Além da interação, cada agente deve ser capaz de aprender e cooperar no ambiente. Em ambientes complexos, um único agente só pode, ao longo do tempo, adquirir experiências suficientes que convergem para uma política ótima, se e somente se uma grande quantidade de episódios é possível, bem como se estratégias específicas são utilizadas para evitar máximos locais. No entanto, em um sistema com vários agentes, valores contraditórios para recompensas acumuladas podem ser gerados, à medida que cada agente utiliza apenas valores locais de aprendizagem. Dessa forma, a aprendizagem coletiva, diferentemente da aprendizagem com um único agente, pressupõe que o conhecimento relevante ocorre quando recompensas são compartilhadas, intensificando a relação entre os agentes. Uma das propostas deste trabalho é mostrar que a interação entre os agentes melhoram a utilidade da política quando as recompensas são compartilhadas, favorecendo o modelo de aprendizagem por reforço.

2.3.2 Coordenação por Sincronização

A coordenação por sincronização é a maneira mais simples e limitada de coordenação, a qual deve descrever precisamente a sequência de ações concorrentes. A sincronização pode gerar uma simultaneidade de várias ações e verificar se os resultados das operações são coe-

rentes. Desta forma, é necessário definir a relação de tempo existente entre as ações de modo que sejam executadas na ordem correta e produzam o resultado esperado (Ferber, 1999).

Geralmente, quando diversos agentes têm acesso e compartilham o mesmo recurso, suas ações precisam ser sincronizadas, de forma que o recurso não fique escasso, evitando conflitos e incoerências (Ferber, 1999). Como exemplo desse tipo de coordenação, podemos considerar o fato de andar de bicicleta, onde existem dois agentes (ação esquerda e ação direita). Para que ocorra o processo de pedalar de maneira sincronizada, é necessário que ambos coordenem suas ações, quando um pressionar o pedal, o outro deve relaxar e vice-versa. A coordenação das ações de um ciclista pode ser representada de uma forma bastante simples como ilustrado na figura 2.4. Cada etapa é representada sob a forma de uma localização e uma transição, com o local que representa a ação na posição superior e a transição corresponde à ação de pressionar o pedal.

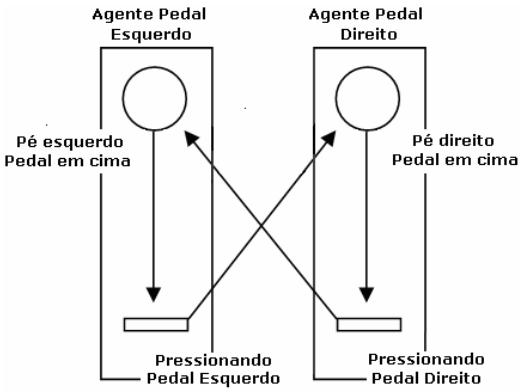


Figura 2.4: Sincronização de ações (Ferber, 1999)

Outro exemplo de sincronização ocorre quando dois robôs são responsáveis pela montagem de peças em uma fábrica, onde a máquina só aceita um único robô por vez. As ações de um robô irão obviamente afetar as ações do outro. Desta forma surge a necessidade de introduzir mecanismos de sincronização, para que o segundo robô não prejudique o trabalho do primeiro robô, quando o mesmo estiver operando a máquina. Para resolver esse problema, os robôs precisam se organizar de uma forma que um espere o outro terminar de montar a sua peça, para só então começar a realizar o seu trabalho. Para resolver tal problema utilizando agentes inteligentes (consideramos também a máquina como um agente), mensagens podem ser utilizadas informando se a máquina está disponível ou não, assim como os semáforos de um sistema operacional multitarefa garantem a execução de vários processos em um mesmo

processador. No entanto, os robôs devem se coordenar para que a ordem de utilização seja respeitada (Ferber, 1999).

2.3.3 Coordenação por Planejamento

A coordenação por planejamento é um método tradicional em sistemas multiagente. O método é dividido em fases. Na primeira, é determinado um conjunto de ações a serem realizadas para atingir o objetivo global, ocorrendo a elaboração de planos. Já na segunda, os planos são selecionados e na sequência executados. Os planos escolhidos podem ser revisados durante a sua execução. Os diferentes planos elaborados pelos agentes podem ocasionar conflitos de objetivos ou de acesso a recursos. Portanto, os planos devem ser coordenados de forma a resolver os conflitos e satisfazer os objetivos dos agentes (Ferber, 1999).

Em sistemas multiagente, o planejamento pode ser divido em três fases: elaboração de planos; sincronização/coordenação; e execução de planos. Durfee (1999) apresenta os seguintes modelos de coordenação distribuídos para um sistema multiagente:

- Planejamento centralizado para planos distribuídos: apenas um agente planeja e organiza as ações para todos os agentes que irão apenas executar os planos. Nessa etapa o agente coordenador terá a visão global do sistema e pode definir as relações de coordenação;
- Coordenação centralizada para planos parciais: onde apenas a etapa de coordenação é centralizada, cabendo a cada agente a função de desenvolver seus próprios planos parciais, e encaminhá-los para o coordenador, o qual é responsável em avaliar tais ações classificando-as de forma que elimine possíveis conflitos; e
- Coordenação distribuída para planos parciais: não existe um coordenador central. Desta forma, cada agente planeja individualmente as ações que deseja executar de acordo com seus objetivos. Para que agentes possam trocar informações sobre seus planos, foi desenvolvido por Durfee e Lesser (1991) o planejamento parcial global.

Grosz e Kraus (1996) desenvolveram um modelo de planejamento colaborativo, denominado *sharedplans*. O modelo fornece uma especificação para projetar agentes com capacidades de colaboração e um framework para identificar e investigar questões sobre colaboração. Hadad e Kraus (1999) apresentam vários exemplos ilustrativos que usam o *sharedplans* para melhorar o uso dos recursos, a coordenação das tarefas e aumentar a utilidade dos agentes para alcançarem os objetivos propostos usando o planejamento. As propriedades particulares do *sharedplans* contribuem com tais melhorias fornecendo aos agentes a possibilidade de

planejar e agir, incluindo comportamentos que podem levar os agentes a comportar-se adequadamente e restrições que proíbem a adoção de intenções conflitantes. Grosz *et al.* (1999) descrevem algumas técnicas para o planejamento colaborativo e sistemas para a comunicação homem-máquina baseado no modelo *sharedplans*. Por exemplo, (i) o GigAgents é um sistema multiagente para a colaboração de grupos heterogêneos de pessoas e agentes; (ii) o webTrader é um sistema multiagente que atua como um ambiente colaborativo no comércio eletrônico e; (iii) o DIAL é um sistema que fornece uma interface colaborativa para a aprendizagem a distância. Essas técnicas motivaram o desenvolvimento de ambientes, onde vários sistemas e pessoas conseguem colaborar usando as técnicas de planejamento colaborativo.

2.3.3.1 Planejamento Global Parcial

O planejamento global parcial proposto por Durfee e Lesser (1991) tem por objetivo aumentar a qualidade da coordenação dos agentes, evitando que realizem atividades redundantes ou se tornem ociosos, auxiliando na organização do sistema multiagente.

O planejamento global parcial é uma técnica desenvolvida para controlar sistemas distribuídos, visando à resolução coerente de um dado problema. Trata-se de uma abordagem flexível de coordenação, que não assume qualquer distribuição de subproblemas, mas permite que agentes se coordenem em resposta à situação atual (Durfee e Lesser, 1991). Cada agente pode representar e raciocinar sobre as ações e interações do grupo de agentes e como essas ações afetam as atividades locais do sistema. Essas representações são chamadas de planos globais parciais. Cada agente pode manter seu próprio conjunto de planos, podendo ser utilizado independentemente ou assincronamente para coordenar suas atividades aos demais agentes de um sistema.

Um planejamento global parcial contém um (i) objetivo, que contém informações do planejamento global parcial, inclui metas, sua importância em forma de grau de prioridade ou razões para sua utilização; (ii) um mapa de atividades, que representa as atividades dos agentes (o que estão executando), incluindo um descritivo dos planos mais relevantes, custos e resultados esperados; (iii) um grafo de construção da solução, que contém informações sobre como os agentes devem interagir, incluindo especificações de quando e quais resultados parciais devem ser trocados com os demais agentes; e (iv) o acompanhamento de todo o processo, que requer informações a serem registradas no planejamento global parcial, incluindo ponteiros para os dados mais relevantes recebidos de outros agentes e quando foram recebidos.

O planejamento global parcial é uma estrutura geral para representar as atividades coordenadas em termos de objetivos, interações e relações entre os agentes, a qual conta com um

agente denominado de *pgplanner*, responsável por verificar o estado atual da representação dos objetivos, ações e planos dos demais agentes que constituem o sistema multiagente. Assim, buscam reunir as atividades comuns, relacionando e reorganizando ações no intuito de alcançar objetivos maiores (Durfee e Lesser, 1991).

O *pgplanner* desenvolve e mantém atualizado o mapa de atividades, permitindo a distribuição de tarefas entre os agentes de forma cooperativa. Isso é necessário para que as soluções parciais das ações de um agente, beneficiem os demais agentes na resolução de seus próprios subproblemas. O *pgplanner* é responsável também pela manutenção do grafo de construção da solução, atualizando de forma constante e reordenando as atividades dos agentes, para então identificar quando e onde o resultado parcial deve ser compartilhado entre os agentes. Isso auxilia o agente a concluir sua tarefa de maneira mais eficiente.

O planejamento global parcial foi utilizado num dos primeiros simuladores para testes de um sistema multiagente: a bancada de monitoramento distribuída de veículos. Este simulador utilizou o conceito de resolução distribuída de problemas, sob o domínio da detecção e monitoramento distribuído de veículos, o qual consistia em detectar e seguir um conjunto de veículos que passavam por uma determinada região, monitorados por um conjunto de sensores distribuídos (Durfee e Lesser, 1991).

O domínio da aplicação era especialmente apropriado para agentes com capacidade de perceber o ambiente e responder as mudanças, pois sempre que um novo veículo era observado, o sistema disparava um processo de detecção e seguimento. A rapidez do processamento era fundamental, uma vez que o domínio era dinâmico e exigia que os agentes determinassem as trajetórias dos veículos presentes em tempo real.

2.3.3.2 Planejamento Parcial Global Generalizado

O planejamento parcial global generalizado proposto por (Decker e Lesser, 1992), é um conjunto de mecanismos de coordenação que atuam unidos a uma arquitetura de agente e a um escalonador de tarefas local, para que agentes possam comunicar e planejar suas ações. Em relação à arquitetura do agente, este possui um conjunto de crenças sobre as tarefas a serem executadas, onde cada agente possui um escalonador de tarefas local. A função do mecanismo de coordenação é fornecer informações ao agente, para que as tarefas sejam realizadas de maneira adequada.

O planejamento parcial global generalizado visa abstrair a coordenação do sistema, separando os processos de coordenação da programação em geral e foi estendido por (Decker e Lesser, 1995) com cinco novas metodologias:

- Comunicação de informação: para a atualização de perspectivas não-locais, de forma que um agente partilhe informações relacionadas à sua visão local com outros agentes, fazendo com que estes atinjam uma visão mais completa do ambiente;
- Comunicação de resultados: a troca de resultados obtidos entre agentes pode beneficiar de forma a tornar mais eficiente a solução de problemas ainda não resolvidos;
- Tratamento de redundância: a redundância pode ser deliberada, preocupando-se com a confiabilidade dos resultados obtidos entre dois ou mais agentes, porém, em geral, ela representa um desperdício de recursos e deve ser evitada;
- Tratamento de relações rígidas de coordenação: uma determinada ação de um agente pode interferir diretamente em uma ação executada por outro agente. Este problema pode ser evitado através do reescalonamento de ações conflituosas; e
- Tratamento de relações flexíveis de coordenação: nesta forma o reescalonamento não é obrigatório, porém pode causar interferência na eficiência ou qualidade de execução das atividades realizadas pelos agentes.

Em comparação com o planejamento parcial global, o planejamento parcial global generalizado acrescenta ainda o escalonamento de tarefas com *deadlines*, heterogeneidade nos agentes e comunicação a múltiplos níveis de abstração. Estes adicionais tornam este mecanismo mais flexível e utilizável na prática. O mecanismo foi implementado no simulador TAEMS (*Task Analysis, Environment Modeling and Simulation*) (Decker e Lesser, 1993). Este último é um simulador para ambientes multiagente que mostra graficamente as tarefas, os dados estatísticos e as ações dos agentes presentes no sistema, permitindo modelar um tipo de ambiente com tarefas computacionais complexas, nas quais podem ser utilizadas abordagens baseadas em agentes. O sistema é dividido em três camadas:

i) nível objetivo: descreve a estrutura essencial do ambiente e suas tarefas;

ii) nível subjetivo: descreve a forma que os agentes percebem e atuam sobre o ambi ente e;

iii) nível generativo: são descritas as características necessárias para gerar a informa ção objetiva e subjetiva em um dado domínio (Decker e Lesser, 2003).

Neste contexto, outro modelo de coordenação proposto por (Tambe, 1997), chamado de STEAM (*Simply, a Shell for Teamwork*), é baseado na teoria de intenções conjuntas e na

teoria de planos conjuntos, onde consistem em coordenar agentes cujos objetivos são idênticos, e que trabalham formando uma equipe para atingir o objetivo de maneira eficaz.

2.3.4 Coordenação Reativa

A coordenação reativa consiste na reação do agente a modificações que ocorrem em seu ambiente e na adaptação de suas ações em relação às ações dos demais agentes (Ferber, 1999). A técnica se torna mais adequada em situações na qual é difícil prever os estados futuros do ambiente, possibilitando as ações sem haver o planejamento antecipado, onde a reação dos agentes depende apenas da percepção do ambiente (Arkin, 1990).

Brooks (1986) propôs a arquitetura *subsumption* empregada no contexto da robótica móvel. Tal arquitetura difere de abordagens tradicionais, onde para construir um sistema de controle de robôs, é realizada a divisão do controle em unidades funcionais. Dessa forma, cada unidade está conectada a níveis vizinhos, de maneira que o sistema seja projetado de forma completa, já que uma unidade individualmente não conseguiria realizar todas as atividades. Portanto, quando há necessidade de inserir novas atividades, todo o projeto é alterado e o controle é dividido em módulos funcionais como percepção, modelagem, planejamento, execução da tarefa e controle dos motores (figura 2.5).

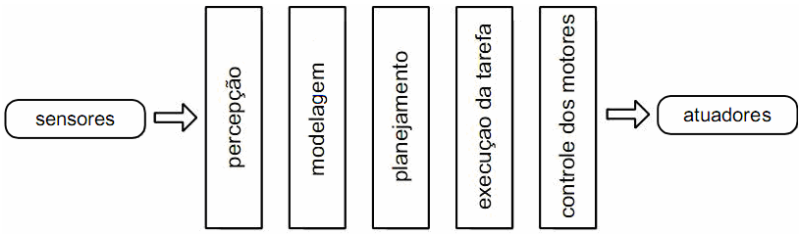


Figura 2.5: Divisão tradicional do sistema de controle em módulos funcionais (Brooks, 1990)

Na arquitetura *subsumption*, o agente é organizado como um conjunto de camadas que representam tarefas ou comportamentos completos (figura 2.6). Além disso, alguns comportamentos podem ativar ou inibir o comportamento de camadas inferiores. Neste caso, o agente opera em nível baixo de abstração, sem ter conhecimento prévio do ambiente, estabelecendo um raciocínio lógico complexo baseado no princípio da reatividade dos agentes e na interação entre os comportamentos locais.

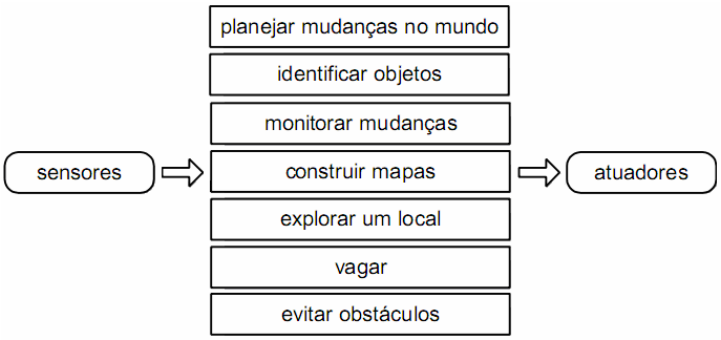


Figura 2.6: Divisão do sistema em camadas de tarefas (Brooks, 1990)

A divisão em camadas de atividades permite acrescentar quando necessário um comportamento, gerando uma nova camada. A intenção é construir um sistema autônomo e simples, podendo ser testado em ambientes do mundo real.

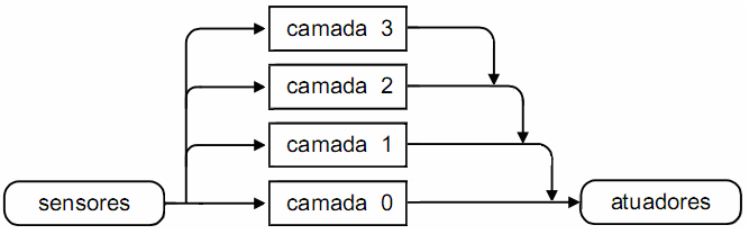


Figura 2.7: O sistema pode ser particionado em qualquer nível, e as camadas abaixo formam um completo sistema de controle (Brooks, 1990)

Na figura 2.7, os níveis elevados suprimem o fluxo de dados das camadas inferiores. O sistema pode ser separado em qualquer nível e as camadas inferiores irão continuar formando o sistema. Devido a essa estrutura, a arquitetura é denominada de *subsumption*.

Brooks (1990) implementou inicialmente a arquitetura *subsumption* em um robô que possuía três camadas de comportamentos e um conjunto de sonares, os quais dão a medida de profundidade a cada tempo. A camada de nível mais baixo é a camada zero, que é responsável por evitar a colisão com obstáculos. A segunda camada, camada um, faz o robô parar quando não está evitando obstáculos e a camada dois incrementa a capacidade de fazer o robô explorar.

Mahadevan e Connel (1992) utilizaram a arquitetura *subsumption* em um robô com um sistema de controle central, com um número de pequenos processos paralelos e concorrentes. Cada um desses comportamentos usa um subconjunto de dados avaliados pelos sensores para controlar os parâmetros de saída. Nesta implementação cada camada de controle consiste de um módulo, que gera um comportamento específico no robô. Um módulo tem dois componen-

tes internos: o bloco política, indica o que fazer com a informação sensorial e o bloco predicado de aplicabilidade, informa quando realizar a ação. Para construir um sistema de controle usando a arquitetura *subsumption*, as tarefas globais são divididas inicialmente em subtarefas. Na sequência, para cada subtarefa, dispositivos são projetados com planos de geração de ação e condições de aplicabilidade. Em seguida as ordens de prioridade dos comportamentos são definidas permitindo ao sistema resolver alguns conflitos que possam existir entre as camadas.

Outro método de controle e resolução de problemas foi proposto por Ferber (1999), onde o problema é decomposto em um conjunto denominado eco-agentes. Na resolução eco-problema, cada eco-agente possui um objetivo a atingir um estado de satisfação e dois comportamentos gerais que são: de satisfação, procura atingir seu estado de satisfação e de fuga, do agente que está agredindo. Um eco-agente apresenta quatro estados internos: i) satisfeito: estado alcançado quando o agente atinge seu objetivo, não sendo necessária outra ação do agente. Porém, quando atacado, o agente altera seu estado para busca de um local para fugir; ii) busca por satisfação: é o estado inicial do eco-agente. Neste estado, o agente realiza ações a fim de alcançar seu objetivo. Um agente em busca por satisfação pode mudar tanto para satisfeito (quando encontra um objetivo que o satisfaça) como para busca de um local para fugir (quando outro agente o ataca); iii) busca de um local para fugir: neste estado o agente atacado procura por um local para escapar do ataque. Após encontrar um local de fuga, o agente altera seu estado para fuga; e iv) fuga: ao encontrar um local para fugir, o agente deve realizar a fuga. Após isso, o agente agressor deve retirar o ataque e o agente agredido volta ao estado em busca por satisfação.

O diagrama da figura 2.8 mostra os estados de um eco-agente e as possíveis transições entre eles. Cada mudança de um estado a outro corresponde às ações dos agentes (Ferber, 1999).

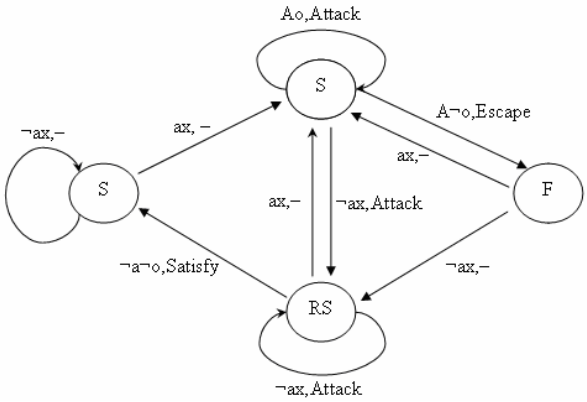


Figura 2.8: Diagrama do eco-agente (Ferber, 1999)

2.3.5 Coordenação por Formação de Coalizão

Em ambientes onde agentes com habilidades diferentes estão inseridos, a convergência pode ser demorada ou não ocorrer. Desta forma surge uma proposta interessante, formar grupos de agentes (coalizão) com interesses em comum, que cooperam e compartilham conhecimentos para reduzir custos e atingem seus objetivos rapidamente, quando comparado com ações individuais (Sandholm e Lesser, 1995).

Uma coalizão é um grupo de agentes que decidem cooperar, a fim de executar uma tarefa comum (Shehory e Kraus, 1995). Os agentes podem determinar a importância das tarefas a executar e participar de mais de uma coalizão. Agentes membros de uma coalizão recebem uma recompensa quando satisfazem a tarefa solicitada, onde normalmente a população de agentes não se altera durante a formação da coalizão.

Uma vez que o grupo de agentes apresenta interesse em comum, suas funções no ambiente são melhor exploradas através da coalizão. Sichman (2003) define coalizão como uma organização de agentes que cooperam para resolver um determinado problema, onde as organizações podem ser classificadas da seguinte maneira:

- Estática: o projetista do sistema tem o total controle sobre a especificação dos agentes e a definição da organização, visando construir um grupo capaz de resolver o problema proposto; e
- Dinâmica: todo o processo de organização deste grupo de agentes ocorre de forma dinâmica, os agentes não possuem papéis pré-definidos e suas funções podem variar.

A formação de coalizão é um processo que deve aumentar a eficiência e adaptabilidade às alterações no ambiente, se exploradas as capacidades dos demais agentes que compõem o

grupo. Entretanto, este processo acaba formando dependências entre os agentes, o que dificulta a alteração de planos individuais (Mérida-Campos e Willmott, 2004).

Quando uma coalizão é formada, os agentes devem se coordenar distribuindo tarefas e sincronizando suas ações, de modo que suas atividades sejam realizadas pouco a pouco, até alcançar o objetivo global do grupo. Assim que a organização de agentes tenha alcançado a solução do problema proposto, esta formação é desfeita, e os integrantes desta estarão disponíveis para participar de outros grupos (Sichman, 2003). Esse autor propôs um modelo de coalizões baseadas em dependência, o qual visa formar organizações de forma dinâmica, baseado na teoria do poder social, que utiliza o conceito de relações de dependência. Neste modelo, antes que os agentes possam iniciar suas atividades dentro de uma coalizão, precisam realizar uma espécie de apresentação, para que tanto o novo integrante, quanto os demais componentes do grupo conheçam as habilidades que cada agente possui. O mesmo ocorre quando um agente decide sair de uma sociedade.

Um dos problemas com técnicas de formação de coalizão está relacionado ao número exponencial de coalizões candidatas. Abdallah e Lesser (2006) desenvolveram um algoritmo distribuído que retorna uma solução em tempo polinomial, garantindo a qualidade desse retorno e aumentando o ganho dos agentes. A solução utiliza fundamentos de organização para guiar o processo da formação da coalizão. Para isso, são usadas técnicas de aprendizagem por reforço para otimizar as decisões de alocação local feita pelos agentes da organização. Na definição do problema, o tempo de uma tarefa é dividido em episódios. Ao iniciar cada episódio, cada agente recebe uma sequência de tarefas. Assim que uma tarefa é alocada a uma coalizão, os agentes pertencentes à coalizão não podem ser alocados para outras tarefas até o final do episódio. Ao final do episódio, os agentes são liberados e podem então ser alocados para a próxima sequência de tarefas. Resultados experimentais mostram o potencial da técnica, verificando a escalabilidade e o número de troca de mensagens.

Mérida-Campos e Willmott (2004) utilizam formação de coalizão que combina fundamentos da teoria dos jogos para cobrir casos onde a população de agentes deve resolver problemas dinâmicos. O método pode levar a uma sequência iterativa de coalizões, sendo que os resultados experimentais mostram como coalizões fortes podem surgir ao longo o tempo, mesmo com estratégias simples. Além disso, a utilidade das coalizões é proporcional ao valor de centralidade e relevância de seus membros para a comunidade.

2.3.6 Otimização Distribuída de Restrição para Coordenação de Sistemas Multiagente

A otimização distribuída de restrição está baseada em técnicas de coordenação que vão além da busca por soluções satisfatórias ou de simples métodos de otimização (Lesser *et al.* 2003). Em otimização por restrição, cada restrição do problema é caracterizada como uma função de otimização (ou função de custo). Desta forma, o mecanismo de busca em um problema de otimização distribuída de restrição preocupa-se em encontrar valores para as variáveis de modo a otimizar as funções de custo, proporcionando garantia de qualidade para as soluções encontradas (Lesser *et al.* 2003).

Um problema de otimização distribuída de restrição é composto por n variáveis V= $\{v_1,v_2,...,v_n\}$, no qual cada variável está associada a um agente x_i. Por sua vez, uma variável contém um domínio finito e discreto, $D_1,D_2,...,D_n$, respectivamente. Apenas o agente x_i é capaz de atribuir valores para a variável v_i e conhecer o domínio D_i. Cada agente deve escolher um valor d_i para sua variável, tal que $v_i \in D_1$. Portanto, a coordenação deve permitir a escolha dos valores para as variáveis de modo a minimizar uma dada função objetivo global definida para o problema (Modi *et al.* 2005).

O conceito de restrição no problema da otimização distribuída de restrição é denominado como função de custo. A função de custo para um par de variáveis x_i e x_j é dada por f_{ij}:D_i x $D_j \rightarrow$ N. Dois agentes x_i e x_j são vizinhos em um grafo de restrições quando existir alguma restrição entre eles. Deste modo, o problema da otimização distribuída de restrição deve então encontrar um conjunto: $A^* = \{d_1, d_2, ..., d_n \mid d_1 \in D_1, d_2 \in D_2, ..., d_n \in D_n\}$ de atribuições para as variáveis, de modo que o custo F acumulado seja mínimo (Mailler e Lesser, 2004). A função objetivo global F é definida na equação 2.1:

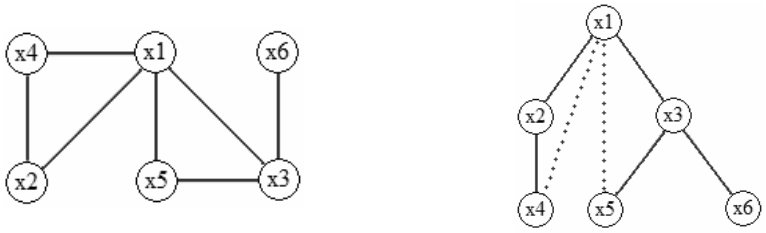
$$F(A) = \sum_{x_i, x_j \in V} f_{ij}(d_i, d_j), onde\ x_i \leftarrow d_i, x_j \leftarrow d_j, em A \qquad (2.1)$$

Os métodos para resolução da otimização distribuída de restrição podem ser divididos em duas categorias: síncronos e assíncronos. Entretanto, os métodos síncronos são dispendiosos, no sentido que os agentes devem aguardar até o recebimento de uma mensagem particular para continuar a processar (Modi *et al.* 2005). Tal característica onera o desempenho da busca, devido ao fato de não ser possível explorar as vantagens do processamento paralelo ao distribuir o problema. Em contrapartida, em métodos assíncronos os agentes devem ser capazes de tomar suas ações com base em suas visões locais do problema, o que aumenta a complexidade

do mecanismo de busca. Em função do desempenho dos métodos síncronos, esta subseção aborda com maior ênfase os algoritmos de busca assíncrona.

Um dos principais algoritmos para resolução da otimização distribuída de restrição é o *Asynchronous Distributed Constraint Optimization* (ADOPT), proposto por (Modi *et al.* 2005). O ADOPT foi o primeiro algoritmo assíncrono completo a oferecer garantia de qualidade aliado a um método de busca assíncrono. Portanto, o ADOPT é capaz de encontrar soluções ótimas usando comunicação assíncrona e localizada entre os agentes (apenas entre os agentes vizinhos).

No ADOPT os agentes devem ser priorizados em uma estrutura de pseudo-árvore. Por meio desta ordem de prioridade, o ADOPT executa uma busca em profundidade por *backtracking* distribuído usando uma estratégia oportunista, isto é, cada agente mantém a escolha do melhor valor baseado em sua visão local. Deste modo, uma rotina de pré-processamento é necessária para transformar o grafo de restrições do problema em uma estrutura de pseudo-árvore. A figura 2.9 ilustra um exemplo de uma pseudo-árvore gerada a partir de um grafo de restrições.



Grafo de restrições (a) Pseudo-árvore (b)

Figura 2.9: Pseudo-árvore gerada a partir de um grafo de restrições (Modi *et al.* 2005)

O grafo apresentado na figura 2.9 é cíclico. Uma das alternativas para eliminar os ciclos do grafo e consequentemente facilitar o processo de busca é transformá-lo em uma pseudo-árvore. Por definição, uma pseudo-árvore é semelhante a uma árvore tradicional, porém, cada nó pode estar conectado a múltiplos nós de maior hierarquia. Contudo, apenas um dos nós de maior hierarquia é definido como pai, enquanto os demais nós de maior hierarquia são denominados pseudo-pais. A figura 3.9 (b) ilustra o exemplo de uma pseudo-árvore, onde as linhas contínuas representam as ligações entre pai e filho e as linhas pontilhadas representam as ligações entre pseudo-pai e pseudo-filho. Maiores detalhes do algoritmo ADOPT podem ser encontrados em (Modi *et al.* 2005). Além disso, uma grande variedade de algoritmos para problemas da otimização distribuída de restrição foram propostas. Dentre estes podemos citar

Dynamic Programming OPtimization (DPOP) e suas variantes (Petcu e Faltings, 2005) e outros algoritmos como o *Optimal Asynchronous Partial Overlay* (OptAPO) (Mailler e Lesser, 2004) e o *No-Commitment Branch and Bound* (NCBB) (Chechetka e Sycara, 2006).

Além dos métodos descritos anteriormente, é possível encontrar na literatura outros métodos menos explorados de coordenação com diferentes abordagens. Dentre estes podemos citar:

- **Coordenação *Look-Ahead***: visa aumentar a visibilidade global dos agentes e fornecer informações para tomada de decisões, já que os agentes necessitam coordenar suas ações constantemente, a fim de completar suas tarefas e melhorar o desempenho do sistema. O escalonamento das operações é realizado pelos agentes através de um algoritmo simples baseado em regras de prioridade, que indicam as operações que esperam por execução. Cada tarefa é completada pela execução das operações pelos agentes. Esse método foi usado por (Liu e Sycara, 2001);
- **Coordenação por Regulamentação**: é um método baseado em leis ou convenções sociais utilizadas para assegurar a coordenação imediata. O princípio deste método é utilizar regras de comportamento que visam eliminar possíveis conflitos. Este é um método raramente descrito na literatura, mas é frequentemente posto em prática em sistemas que exigem coordenação limitada como modelos macroscópicos de simulação. O princípio deste método é utilizar regras de comportamento que visam eliminar possíveis conflitos. Por exemplo, atribuir regras de prioridade á veículos em cruzamentos, com o objetivo de evitar colisões (Ferber, 1999);
- **Coordenação por Pontos Focais**: coordenação baseada em interações humanas livre de comunicação, abordando os pontos focais como uma heurística para a coordenação em ambientes reais (Fenster e Kraus, 1998). Um exemplo de sucesso na aplicação desta abordagem é a coordenação de escolhas comuns entre agentes em simulações (dois agentes escolherem o mesmo objeto em um ambiente sem comunicação). Algoritmos de pontos focais são capazes de identificar, em um ambiente, objetos com propriedades diferentes dos demais e de fornecer formas de escolha destes objetos pelos agentes; e
- **Coordenação por Matriz de Possibilidades**: a coordenação requer que um agente reconheça o estado corrente do ambiente e modele as ações dos outros agentes para decidir seu próprio comportamento (Noh e Gmyrasiewicz, 1997). Nesta abordagem, cada

agente é independente para tomar decisões e executar suas ações, sendo que a coordenação entre os agentes emerge como resultado das ações dos agentes individuais. Não há comunicação entre os agentes e é utilizado o método de modelagem recursiva (Gmytrasiewicz e Durfee, 1995).

Na próxima seção, é apresentada uma avaliação dos principais métodos de coordenação a partir de critérios e abordagens encontradas na literatura.

2.4 Critérios de Análise e Comparação para Coordenação

Além das técnicas de coordenação citadas, há ainda diversas outras técnicas de coordenação para sistemas multiagente citadas por (Jennings, 1996; Nwana *et al.* 1996; Ossowski 1999; Ferber, 1999) menos estudadas. Comparar técnicas de coordenação é uma tarefa complexa, devido à quantidade de critérios que devem ser considerados ao longo do processo da análise. Os seguintes critérios podem ser utilizados neste sentido (Frozza e Alvares, 2002):

- Preditividade: capacidade de determinar o estado futuro do ambiente e dos agentes;
- Adaptabilidade: capacidade de adaptar-se a eventos ou a situações inesperados;
- Controle das ações: centralizado ou distribuído;
- Modo de comunicação: forma dos agentes tomarem conhecimento das ações dos outros agentes. Pode ser via interação, percepção, sem comunicação direta ou com comunicação direta;
- Tipo de troca de informação: informação manipulada e trocada entre os agentes para que se efetue a coordenação; útil para aplicações que tratam da elaboração de planos de ação;
- Aplicações a que se destinam: adaptáveis a qualquer domínio (característica que tende a ser menos eficaz) ou adaptáveis a certos domínios específicos;
- Vantagens da abordagem de coordenação utilizada;
- Desvantagens da abordagem de coordenação utilizada;
- Escalabilidade: quantidade de esforço computacional necessário à medida que a complexidade do ambiente aumenta.

Esses critérios contribuíram para a escolha dos métodos de coordenação empregados no domínio da aplicação do trabalho em questão. Em função das características e do objetivo

da aplicação a ser desenvolvida, uma análise das questões que envolvem a coordenação pode contribuir para melhorar o desempenho dos agentes durante a resolução das tarefas.

Os itens I ao IV apresentam a avaliação dos principais métodos de coordenação descritos neste trabalho, considerando os critérios apresentados nesta seção.

I. Formação de Coalizão

Avaliação dos métodos por planejamento e sincronização.

- **Vantagens**: através da formação de grupos de agentes com interesses em comum, é possível combinar as capacidades complementares de cada agente, pois estarão atuando de forma conjunta em busca de um objetivo específico. Portanto, há o aumento da eficiência de execução em tarefas de grupo. Utiliza ideia de utilidade para agentes e para as coalizões.
- **Desvantagens**: um agente tende a ficar dependente de vários outros agentes presentes em seu grupo. Desta forma, suas ações estariam comprometidas e a alteração de planos individuais se torna uma opção pouco viável, pois pode afetar todo o andamento do grupo. Ademais, o número de possíveis coalizões entre agentes cresce de forma exponencial, necessitando de alguma estratégia de controle.
- **Escalabilidade**: a quantidade de dados pode ser bem elevada, tornando viável a presença de diversos agentes com características distintas, distribuídos entre vários grupos formados por indivíduos com interesses em comum. A quantidade de informação transmitida entre indivíduos e a quantidade de coalizões candidatas pode inviabilizar o sistema para um número elevado de agentes.
- **Preditividade**: não ocorre.
- **Adaptabilidade**: os agentes podem se associar a diferentes grupos de uma forma dinâmica. Desde que estes grupos tenham indivíduos em busca de um objetivo pelo qual as ações deste agente podem ser proveitosas. Os agentes adaptam-se a tarefas que variam constantemente.
- **Controle de ações**: distribuído entre os agentes que formam a coalizão, os quais determinam um tipo de escalonamento de atividades, sincronizando ações deste grupo a fim de alcançar o objetivo global do mesmo.
- **Modo de comunicação**: os agentes trocam informações através de mensagens, as quais devem seguir um padrão que seja adotado por todos os membros da equipe..

- **Troca de informações**: as mensagens podem ser interessantes quando um agente oferece uma informação útil para que outro agente modifique seus planos de forma a alcançar seus objetivos mais rapidamente. Podem trocar informações para elaboração de planos.
- **Aplicações:** ambientes de execução de tarefas.

II. Planejamento

Avaliação do planejamento global parcial e planejamento global parcial generalizado, comparando os métodos de coordenação por planejamento e sincronização.

- **Vantagens:** Através de um planejamento antecipado, as ações executadas pelos agentes tendem a ser mais eficientes, pois desta forma estarão evitando atividades redundantes e estarão sempre ativos, reduzindo consideravelmente o tempo ocioso. Independência de uso dos mecanismos de coordenação.
- **Desvantagens**: em ambientes dinâmicos o custo para a elaboração de planos e a escolha destes pode ser elevado, afetando diretamente no desempenho do sistema.
- **Escalabilidade**: devido ao alto custo para o processamento de planos parciais e globais, a quantidade de agentes deve ser moderada, a fim de evitar gargalos na execução do sistema.
- **Preditividade**: não ocorre.
- **Adaptabilidade**: não aborda.
- **Controle de ações**: distribuído entre os agentes. O *pgplanner* especifica os planos pelos quais cada agente deve executar, tornando o controle de ações centralizado.
- **Modo de comunicação**: podem adotar o modelo de comunicação *blackboard*, onde cada agente deposita suas percepções, resultados obtidos e conhecimento sobre o ambiente, que estará disponível para todos os demais agentes, facilitando na alteração de planos parciais, elevando as chances de obter sucesso ao alcançar um determinado objetivo.
- **Troca de informações**: através da troca de informações, os agentes podem alterar seus planos parciais. Trocam informações para realizar o escalonamento de tarefas.
- **Aplicações:** Diferentes aplicações, que envolvem escalonamento de tarefas.

III. Coordenação Reativa

Avaliação dos métodos que utilizam algoritmos baseados em recompensas e colônia de formigas.

- **Vantagens**: a utilização de agentes reativos se torna interessante em contextos dinâmicos, onde há uma grande dificuldade em antecipar as mudanças no ambiente.
- **Desvantagens**: dificuldades em alcançar tarefas coletivas de longo prazo e com alto nível de abstração. Não tem capacidade de planejar sobre eventos futuros.
- **Escalabilidade**: diversos agentes podem ser necessários para lidar com problemas dinâmicos. Os agentes são geralmente simples e exigem pouco recurso localmente.
- **Preditividade**: não ocorre.
- **Adaptabilidade**: por se tratar de agentes que reagem a alterações no ambiente, eles possuem um alto grau de adaptabilidade.
- **Controle de ações**: o controle de ações é totalmente baseado em suas percepções locais sobre o problema.
- **Modo de comunicação**: através de marcações que os agentes podem deixar no ambiente, permitindo aos demais agentes utilizá-las como referência para encontrar mais facilmente a resposta para evitar conflitos e melhorar a utilização dos recursos.
- **Troca de informações**: não há troca de informações.
- **Aplicações**: diferentes aplicações, que envolvem ambientes dinâmicos cuja preditividade seja praticamente inexistente.

IV. Otimização Distribuída de Restrição

Avaliação dos métodos de coordenação reativa e regulamentação.

- **Vantagens**: através da eliminação de resultados menos eficientes, em conjunto com as restrições impostas é possível traçar um conjunto de possíveis soluções ótimas para o problema, e desta forma tornar a busca pelo melhor resultado mais fácil.
- **Desvantagens**: a falta de mecanismos de preditividade e a dificuldade de modelagem.
- **Escalabilidade**: em ambientes dinâmicos onde a busca pela melhor solução exige que todos os agentes estejam constantemente alterando o valor de suas variáveis, o número de agentes deve ser moderado, para que não ocorram problemas com o desempenho do sistema.
- **Preditividade**: não ocorre.

- **Adaptabilidade**: em contextos dinâmicos, podem ocorrer constantemente variações nos valores de restrições, o que exige que os agentes se adaptem a essas regras.
- **Controle de ações**: ocorre através da eliminação de resultados não-satisfatórios, sempre respeitando as restrições impostas pelo sistema.
- **Modo de comunicação**: alteração de valores atribuídos às variáveis.
- **Troca de informações**: os agentes trocam informações apenas com seus vizinhos (pai e filhos). Algoritmos diferem quanto ao tipo de informação trocada, informação local ou global.
- **Aplicações**: domínios cujo objetivo principal seja a otimização de resultados.

É possível observar que diferentes métodos de coordenação podem ser aplicados em sistemas multiagente. Cada método possui características específicas com relação às características dos agentes, ações, maneira de comunicação, e que podem influenciar no desempenho do método de coordenação utilizado.

Observa-se que os métodos de coordenação podem ser combinados ou associados a outras técnicas na resolução de problemas, com o objetivo de atingirem a eficiência na execução das ações de forma coordenada pelos agentes envolvidos. Após análise, algumas observações foram levantadas:

- Com a abordagem da coordenação sem comunicação, as escolhas das ações a serem executadas pelos agentes podem depender do conhecimento obtido com o ambiente e pelo uso da abordagem de pontos focais;
- Com a abordagem da coordenação com comunicação, as informações trocadas entre os agentes são a base para que a coordenação ocorra de maneira eficiente;
- Poucas formas de coordenação se preocupam apenas com a resolução de conflitos, o que pode influenciar negativamente a atuação dos agentes e a resolução das tarefas; e
- Os métodos e as abordagens de comunicação apresentados não abordam a questão do aprendizado. O aprendizado é uma tendência que pode trazer benefícios para a atuação coordenada dos agentes, principalmente em ambientes dinâmicos, onde o tempo para tomada de decisão de ações a serem executadas pode afetar o desempenho e os resultados do sistema.

2.5 Considerações Finais

Observou-se neste capítulo que não há uma definição universalmente aceita na literatura para o termo agente. Agentes são capazes de atuar de maneira autônoma em um ambiente em comum, adaptando-se às tarefas para os quais foram designados, a fim de satisfazer os objetivos estabelecidos. Os termos autonomia e adaptação são as características mais importantes dos agentes. Autonomia é a capacidade de um agente executar o controle sobre suas próprias ações, e adaptação é a capacidade de melhorar seu comportamento em função de experiências anteriores. Essas características são encontradas em agentes conhecidos como autônomos adaptativos. A aprendizagem de um agente pode ser realizada através de tentativa e erro ao atuar sobre um ambiente. Assim, a fonte de aprendizagem do agente é a própria experiência, cujo objetivo formal é adquirir uma política de ações que maximize a função objetivo.

Foram discutidos diversos métodos de coordenação para agentes e observado que a aplicação de agentes de aprendizagem no problema de coordenação de sistemas multiagente tem se tornado cada vez mais frequente e necessário. Há diferentes critérios para selecionar um método de coordenação e a escolha depende das características e do objetivo da aplicação. Isso ocorre porque a adaptação dos modelos de coordenação geralmente é necessária em problemas complexos, eliminando e/ou reduzindo deficiências dos mecanismos de coordenação tradicionais, tais como escalabilidade, preditividade, comunicação e adaptabilidade. Nessas condições, a coordenação é o ato de gerenciar dependências entre atividades. Estas dependências podem aumentar, como consequência de atividades sendo executadas em um mesmo ambiente, e podem ocorrer naturalmente quando os agentes estão em um ambiente comum e compartilham recursos.

No capítulo seguinte são apresentados os conceitos da teoria das redes sociais e fundamentos matemáticos da teoria dos grafos. Discutimos ainda a relação das redes sociais e dos sistemas multiagente na construção de estruturas sociais. Nesse capítulo será mostrado como esses conceitos podem contribuir para a geração de modelos de coordenação baseados em algoritmos de colônia de formigas ou aprendizagem por reforço que melhoram o comportamento dos agentes ao longo do processo de interação.

Capítulo 3

Teoria das Redes Sociais

Foram apresentados no capítulo 2 diversos métodos de coordenação para sistemas multiagente. No capítulo 4 é mostrado como algoritmos baseados em população constituem uma forma coletiva de coordenação para sistemas multiagente, onde a partir das recompensas sociais os indivíduos melhoram o comportamento do grupo reforçando as relações existentes entre os estados do sistema.

Neste capítulo são apresentados os principais conceitos das redes sociais. As redes sociais fornecem ferramentas que permitem analisar as redes de relacionamentos construídas ao longo do processo de interação dos agentes, identificando indivíduos relevantes e as relações mais frequentes que interferem no processo de coordenação quando recompensas são compartilhadas.

Métodos de inteligência de enxames, aprendizagem por reforço e os modelos de sistemas sociais estão baseados em princípios muitas vezes complementares, possibilitando observar o impacto das relações estabelecidas através da aplicação da teoria das redes sociais na adaptação de métodos de coordenação baseados em recompensas. Acreditamos que com os conceitos da análise das redes sociais, a estrutura social construída a partir das interações pode melhorar a coordenação dos indivíduos de um sistema.

As redes sociais foram inicialmente analisadas pela sociologia, psicologia social e antropologia, onde os atributos observados a partir dos grupos sociais (movimentos sociais, grupos étnicos, grupos de empresas ou nações) eram representados em termos de ligações entre os indivíduos da rede (Freeman, 1996).

O sociólogo Jacob L. Moreno é considerado pioneiro da utilização de redes sociais. No artigo "*Who shall survive*?", publicado em 1934, Moreno propôs os sociogramas e as sociomatrizes, utilizadas para representar o relacionamento entre crianças (quem interagia com

quem) (Moreno, 1978). A antropologia e sociologia utilizaram redes sociais para mapear relações familiares durante estudos de campo, estabelecendo laços quando interagiam.

Granovetter em 1973 diferenciou essas relações como: fortes, ausentes e fracas; mostrando aos sociólogos a importância das relações fracas, devido à sua importância de ligação entre os elementos da rede social que não estão conectados diretamente, originando o conceito de ponte. O elemento que faz a ponte é responsável pelo relacionamento entre os subgrupos da rede, portanto, o elemento ponte está fortemente conectado a um grupo que interage com um elemento de outro grupo (Grosser, 1991).

As metodologias para análises das redes sociais começaram a avançar com o desenvolvimento das ferramentas matemáticas e o uso da computação. Atualmente, é possível encontrar o uso dos conceitos de redes sociais em diversas áreas, como: computação, matemática, física, economia, ciências sociais e da informação, saúde pública, biologia, antropologia, sociologia, psicologia, entre outros, aplicados em diferentes domínios, como a Internet, disseminação de vírus, movimentos sociais, redes de terrorismos, importância dos indivíduos para uma organização, estudos epidemiológicos, relacionamentos, modelos de disseminação e marketing de produtos, *etc*.

3.1 Definições de Redes Sociais

Uma das primeiras definições de redes sociais foi descrita por James Clyde Mitchell em 1969, que definiu as redes sociais como um tipo específico de relação que liga um conjunto de objetos ou acontecimentos (Mitchell, 1969). As redes sociais seriam parte integrante da sociedade humana, e poderiam ser usadas para explicar por que a sociedade funciona de tal maneira.

Uma rede social consiste em um conjunto de nós (atores, indivíduos, elementos, estados)[1] e as ligações (conexões ou laços) entre eles (Wasserman e Faust, 1994). Os nós podem representar pessoas, entidades, organizações, sistemas, elementos computacionais, *etc.*, que podem ser analisados individualmente ou coletivamente, observando a relação entre eles. Um dos objetivos é compreender o impacto social dos nós através de suas conexões, na formação da estrutura da rede. Esse estudo é realizado com as métricas da análise de redes sociais.

Na análise de redes sociais os nós estão relacionados por laços, responsáveis por estabelecer a ligação entre pares ou grupos de nós. Os laços fortes indicam relações consistentes

[1] O termo nó pode ser adequado conforme o domínio da aplicação. Em algoritmos de aprendizagem por reforço e colônia de formigas o termo nó pode ser chamado de estado e a ligação entre eles é chamada de relação.

entre os nós, e laços fracos têm a função de ligar partes da rede que não estão ligadas diretamente pelos laços fortes. Essa conexão dá origem ao conceito de ponte (Granovetter, 1973). Dessa forma, o nó que faz a ponte é o responsável pela ligação entre os subgrupos da rede.

Há várias definições e análises matemáticas que envolvem os elementos que compõe uma rede social, tais como: grafos direcionados, subgrafo, *clique*, *cutpoint*, distância geodésica, tamanho do caminho, ponte, díade, tríade, grau do nó, medidas de centralidade e prestígio. Ao longo deste trabalho, tais elementos são descritos e oportunamente mencionados quanto à utilidade observada para algoritmos de otimização.

Quando ocorre a ligação entre dois e três nós formam-se unidades de análise, denominados de díade e tríade respectivamente. A análise de díades procura identificar se a ligação entre os nós é recíproca. Com as tríades pode-se observar a transitividade, analisando o balanço ou equilíbrio estrutural da rede. Um subgrupo é um subconjunto de nós e possíveis ligações. Um *clique* é um subgrupo no qual cada nó tem ligações com todos os demais, sem haver outros nós que tenham conexões com todos os nós do *clique*. Um grupo é um conjunto finito com todos os nós para os quais os laços foram mensurados.

Redes sociais podem ser formalizadas e analisadas com a teoria dos grafos, onde fundamentações matemáticas são utilizadas para compreender os elementos e métodos. Para permitir a visualização e análises numéricas, há disponível vários sistemas computacionais para análise dos dados das redes sociais. Tais indicações são descritas nas próximas seções.

3.1.1 Classificação das Redes Sociais

Recentemente, novos modelos de redes foram desenvolvidos na tentativa de capturar as propriedades observadas nas redes do mundo real. Esses modelos incluem redes do mundo pequeno (Milgram, 1967) e redes de livre escala (Barabási *et al.* 2000). Além desses modelos, há os grafos regulares e grafos aleatórios, tipicamente usados para estudar os sistemas sociais.

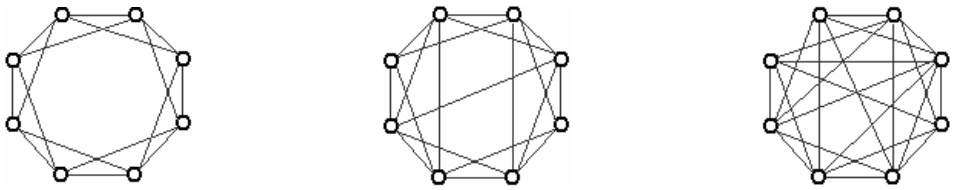
Grafos aleatórios foram inicialmente apresentados por Erdös e Rényi em 1960, mostrando os princípios da formação de redes sociais (Erdös e Rényi, 1960). Um grafo aleatório $G_{N,p}$ consiste de N nós que estão conectados aleatoriamente, onde p denota a probabilidade de existir uma ligação entre um par de nós escolhido aleatoriamente. Grafos aleatórios são amplamente estudados, pois muitas de suas propriedades podem ser analiticamente computadas. Por exemplo, o número médio de ligações não direcionadas em $G_{N,p}$ é $N(N-1)p/2$, e o grau médio do nó é $k=p\,(N-1) \approx pN$.

Erdös e Rényi exemplificam que uma única conexão entre cada um dos convidados de uma festa faria com que todos estivessem conectados ao final da mesma. Assim, quanto mais

conexões fossem adicionadas, maior seria a probabilidade de gerar grupos. Dessa forma, uma festa poderia ser um conjunto de grupos, que estabelecem relações aleatórias com os demais. Para Erdös e Rényi a relação entre os convidados acontece de maneira aleatória, ou seja, o processo de formação dos grafos é aleatório. Assim, concluíram que todos os nós, em uma determinada rede, deveriam ter mais ou menos a mesma quantidade de conexões, ou a mesma probabilidade de receber novas conexões, constituindo-se como redes igualitárias (Barabási, 2003a).

O modelo de rede de *mundos pequenos* é uma tentativa de introduzir mais agrupamentos na rede e computar o comprimento médio dos caminhos (Watts e Strogatz, 1998). A principal observação é que as redes do mundo pequeno possuem propriedades identificadas em redes regulares e grafos aleatórios. Um exemplo é uma rede na forma de anel, onde cada nó se conecta aos nós mais próximos. A característica chave é que para qualquer vizinhança, a maioria dos nós estará conectada a outros agrupamentos.

Uma maneira para diminuir o comprimento médio do caminho é usar uma probabilidade para ligações aleatórias, resultando em conexões *shortcut* através do grafo, conforme figura 3.1. O parâmetro *p* é usado para determinar se uma ligação é substituída por um *shortcut*. É possível observar que na construção de uma rede mundo pequeno o grafo pode tornar-se desconectado. Quando as arestas são substituídas por *shortcuts* com probabilidade *p*= 1 o grafo será aleatório.



(a) *p*=0 (b) *p*=0,1 (c) *p*=0,3

Figura 3.1: Redes do mundo pequeno, onde: a) rede sem ligações *shortcut*; b) rede com poucas *shortcut*; e c) mundo pequeno com muitos *shortcuts*, semelhante a um grafo quase completo (Gaston e DesJardins, 2005)

O modelo de rede de mundos pequenos foi usado por Milgram na década de 60, na intenção de observar o grau de separação entre as pessoas (Milgram, 1967; Watts, 2003). Para isso, ele enviou de maneira aleatória uma determinada quantidade de cartas para algumas pessoas. A mensagem explicava que a carta deveria ser entregue para uma pessoa específica. Caso não o conhecessem, deveriam então direcionar a carta para uma pessoa que acreditassem conhecer o destinatário. Com o experimento, observou-se que as cartas chegariam ao destina-

tário passando por uma quantidade pequena de pessoas. Isso indicou que as pessoas estariam a poucos graus de separação, o que se denominou de mundo pequeno.

Isso mostra a importância do trabalho de (Granovetter, 1973), mostrando que pessoas com pouco relacionamento, chamados de laços fracos, eram muito mais importantes na manutenção da rede social, do que pessoas com forte relacionamento, chamados de laços fortes, pois conectariam as pessoas a outros grupos sociais. Dessa forma, as redes sociais não são aleatórias, pois existe alguma ordem na formação de sua estrutura (Watts, 2003).

A partir de tais experimentos e teorias, Watts (2003), e Watts e Strogatz (1998), descobriram que as redes sociais apresentavam padrões altamente conectados, tendendo a formar pequenas quantidades de conexões entre cada pessoa. O modelo de Watts e Strogatz é especialmente adaptado às redes sociais e mostram um modelo mais próximo da realidade. Em larga escala, essas conexões mostram a existência de poucos graus de separação entre as pessoas. Eles criaram um modelo semelhante ao de Erdös e Rényi, onde os laços eram estabelecidos entre as pessoas mais próximas e alguns laços estabelecidos de modo aleatório entre algumas pessoas, mostrando uma rede como um mundo pequeno (Watts, 2003).

Apesar de estabelecer certos padrões, Milgram (1967) e mais tarde Watts (1999), assumiam as redes sociais como redes aleatórias, como Erdös e Rényi. Por sua vez, Barabási (2000) demonstrou que as redes não são formadas de modo aleatório, mas que existe uma ordem na dinâmica de estruturação. Este padrão de estruturação foi identificado por Barabási, mostrando que quanto mais conexões um indivíduo possui, maior a probabilidade de ter novas conexões, conceito conhecido como ‘ricos ficam mais ricos’. Isso implicaria que as redes não seriam constituídas de nós igualitários, ou seja, com a possibilidade de haver uma distribuição uniforme do número de conexões. Ao contrário, em tais redes haveria poucos nós com muitas conexões (*hubs*), e muitos nós com poucas conexões. Portanto, os *hubs* seriam os ricos, que tem maior probabilidade de receber mais conexões. Redes com essas características foram denominadas de redes livres de escala (Barabási e Bonabeau, 2003b).

Um modelo de rede livre de escala é motivado pela distribuição de grau da Internet e a WWW (Barabási, 2002), onde há poucos sites com muitas ligações e muitos sites com poucas ligações. O modelo de Barabási segue esse exemplo. Já o modelo de Watts e Strogatz tem um grau de conectividade parecido com os grafos aleatórios de Erdös e Rényi, onde os nós possuem uma quantidade semelhante de ligações. Além disso, os modelos livres de escala são utilizadas para a modelagem de redes que possuem tamanho variável, ou seja, um número indefinido de indivíduos pode ser adicionado na rede gradualmente.

3.2 Fundamentos Matemáticos e a Teoria dos Grafos

Seja V um conjunto finito e não vazio de nós, e E uma relação binária sobre V. O par ordenado $(v,w) \in$ E, (ou simplesmente vw), onde v, $w \in$ V, é representado por uma linha ligando v a w. Tal representação de um conjunto V e a relação binária G=(V,E) sobre o mesmo é denominada grafo (West, 2001).

Os elementos de V são denominados vértices (nós, pontos), e os pares ordenados de E são denominados de arestas (ligações, linhas ou arcos do grafo). Uma aresta é dita incidente com os vértices que ela liga. Uma aresta incidente a um único vértice é denominada um laço. Dois vértices são adjacentes, se eles estão ligados por uma aresta. Um vértice é dito isolado se não existe aresta incidindo sobre ele.

A figura 3.2 mostra uma representação geométrica do grafo G=(V,E) onde v_5 é um vértice isolado e a aresta (v_1, v_1) é um laço.

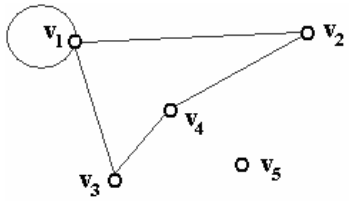
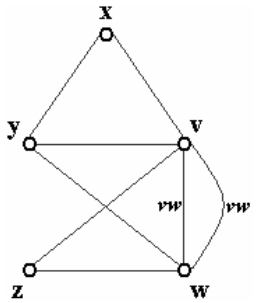


Figura 3.2: V = $\{v_1, v_2, v_3, v_4, v_5\}$ e E = $\{v_1\, v_2, v_1\, v_3, v_2\, v_4, v_3\, v_4, v_1\, v_1\}$

O cardinal |V| = n é a ordem de G, adotando-se |E| = m, sem designação específica.

Duas arestas que incidam sobre o mesmo vértice são ditas adjacentes. Se existem duas arestas $e_i = (v,w)$ e $e_j = (v,w)$, então diz-se que e_i e e_j são arestas paralelas (figura 3.3).



V = $\{x, y, z, v, w\}$

E = $\{xy, xv, yv, yw, vw, vw, zw\}$

Figura 3.3: As arestas dos vértices v e w são paralelas

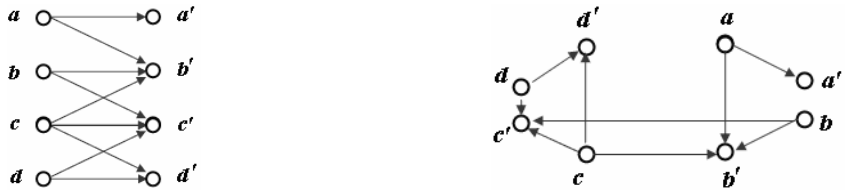
Se um grafo possui arestas paralelas, então este grafo é denominado de multigrafo. Caso contrário, diz-se que o grafo é simples. Um grafo simples, em que cada par distinto de vér-

tices é adjacente, é denominado grafo completo. O grafo completo de *n* vértices é usualmente representado por K_n. Todo grafo completo de *n* vértices possui $m = \left[\frac{n}{2}\right]$ arestas.

Um grafo $\overline{G}$ é dito complementar de G se possui a mesma ordem de G, e se uma aresta $(v_i, v_j) \in G$ então $(v_i, v_j) \notin \overline{G}$. Se $G=(V_1 \cup V_2, E)$ é tal que $V_1 \cap V_2 = \phi$ e toda aresta $(v_i, v_j) \in E$, tem-se que $v_i \in V_1$ e $v_j \in V_2$, então o grafo é denominado grafo bipartite e denotado por $K_{r,s}$, onde $|V_1| = r$ e $|V_2| = s$.

Um grafo é dito dirigido (ou dígrafo), se suas arestas possuem orientação, caso contrário o grafo é não dirigido. Um grafo não dirigido é uma representação de um conjunto e uma representação simétrica binária sobre esse conjunto. Em um grafo não dirigido, uma aresta ligando dois vértices *v* e *w* pode ser representada por (*v*,*w*) ou (*w*,*v*) indistintamente, diferentemente de um dígrafo.

Desde que grafos podem ser usados para representar uma classe muito geral de estruturas, a teoria dos grafos é uma importante área de estudo na matemática combinatória. Por exemplo, considerando a transmissão de quatro mensagens, *a*, *b*, *c* e *d*, através de um canal de comunicação, onde o destinatário receberá as quatro mensagens correspondentes *a*', *b*', *c*' e *d*'. Devido à interferência de ruídos no canal de comunicação, uma mensagem pode chegar errada ao destinatário. A relação entre mensagens transmitidas e recebidas pode ser representada por um dígrafo, conforme figura 3.4 (a). Pode-se observar que *a*' ou *b*' será recebido quando *a* for transmitido, *b*' ou *c*' será recebido quando *b* for transmitido e assim sucessivamente (Rabuske, 1992).



(a) (b)

Figura 3.4: Grafos isomorfos

O grafo da figura 3.4 (a) pode ser desenhado de diferentes formas, sendo uma delas mostrada na figura 3.4 (b). Esses grafos são denominados isomorfos (Roberts, 1984). Dois grafos são isomorfos se for possível fazer coincidir, respectivamente, os vértices de suas representações gráficas, preservando as adjacências das arestas. Formalmente pode-se dizer que $G_1=(V_1, E_1)$ e $G_2=(V_2, E_2)$ são isomorfos se satisfazerem as seguintes condições: i) $|V_1|=|V_2| =$

n; e ii) existe uma função biunívoca $f:V_1 \rightarrow V_2$, tal que $(v,w) \in E_1 \Leftrightarrow (f(v),f(w)) \in E_2 \; \forall \; v,w \in E_1$.

Um grafo G'=(V', E') é um *subgrafo* de G=(V,E), se V' for subconjunto de V e E' um subconjunto de E. A figura 3.5 mostra um exemplo.

Seja G=(V,E) um grafo simples. Define-se grau de um vértice $v \in V$, denotado por gr(v), como sendo o número de arestas incidente a v. Um grafo é dito regular de grau r, se todos seus vértices possuem grau r. Se um grafo é regular de grau zero, então o grafo é dito nulo.

G=(V,E) G'=(V',E')

Figura 3.5: Exemplo de subgrafo

Um vértice que não possui aresta incidente é dito isolado ou vértice de grau zero. Um vértice de grau igual a 1 é dito pendente.

Teorema 1: a soma dos graus dos vértices em um grafo (dirigido ou não) é igual a duas vezes o número de arestas.

Prova: Desde que cada aresta contribui na contagem de um no grau de cada dois vértices com os quais é incidente, então cada aresta é sempre contada duas vezes (equação 3.1):

$$\sum_{i=1}^{n} gr(v_i) = 2m \tag{3.1}$$

Teorema 2: Em qualquer grafo existe sempre um número par de vértices de grau ímpar.

Prova: Suponha que exista um grafo G=(V,E) onde todos os vértices possuam grau ímpar, logo:

$$\sum_{i=1}^{n} gr(v_i) = \begin{cases} \textit{número par, se n for par} \\ \textit{número ímpar, se n for ímpar} \end{cases} \tag{3.2}$$

Pelo teorema 1 a soma dos graus dos vértices é par, portanto *n* obrigatoriamente é par (equação 3.2).

Se cada par de vértices no grafo está ligado por um caminho, isto é, para todos x, y ∈ V(G) existe um caminho (x,y), então o grafo é chamado de conexo. Um grafo é dito rotulado quando seus vértices e/ou arestas são distinguidos uns dos outros por rótulos. Caso contrário o grafo é não rotulado (grafos rotulados estão relacionados com problemas de enumeração combinatória) (Rabuske, 1992).

3.2.1 Ciclos *Hamiltonianos*

Um ciclo *hamiltoniano* em um grafo conexo G é definido como um caminho simples fechado, isto é, passa-se em cada vértice de G exatamente uma vez, exceto naturalmente no vértice inicial que é considerado também vértice terminal. Portanto um ciclo *hamiltoniano* em um grafo de *n* vértices consiste de exatamente *n* arestas (Roberts, 1984).

O comprimento do caminho *hamiltoniano* em um grafo conexo de *n* vértices é *n*–1. Obviamente, nem todo grafo conexo possui um ciclo *hamiltoniano*. Portanto, o problema é que não é possível saber antecipadamente se existe condição necessária e suficiente para que um grafo conexo G possua um ciclo *hamiltoniano*.

Essa questão foi proposta pelo matemático William Rowan Hamilton, em 1859, e considerada insolúvel. O problema de Hamilton parece ser, até agora, mais complexo do que o problema de *Euler* (Wilson, 1996). Porém pode-se afirmar que existem certos tipos de grafos que contêm um ciclo *hamiltoniano*, como, por exemplo, um grafo simples, conexo e completo, de $n > 2$ vértices. Se $n = 2$, então G contêm um caminho *hamiltoniano*. Um dos problemas tratados neste trabalho consiste em um grafo *hamiltoniano*, ilustrado pelo problema do caixeiro viajante.

3.2.2 Teoria dos Grafos na Análise de Redes Sociais

A teoria dos grafos vem sendo empregada em análises de redes sociais devido a sua capacidade de representação e simplicidade. Uma rede pode ser interpretada de diferentes maneiras. Uma boa maneira para identificá-la é como um grafo, composto de nós conectados pelas arestas.

Conforme observado, a teoria dos grafos fornece operações matemáticas a partir das quais muitas propriedades podem ser quantificadas. Essa teoria pode fornecer uma lista de termos para denotar as propriedades da estrutura social, fornecendo um conjunto de conceitos que permite referenciar tais propriedades.

Em redes sociais a representação visual por grafos é denominada de sociograma, criado por Moreno em 1978 para representar a relação social entre estados[2] e as ligações (Moreno, 1978). A figura 3.6 exemplifica um sociograma formado pelas interações dos agentes com algoritmos de colônia de formigas no problema do caixeiro viajante.

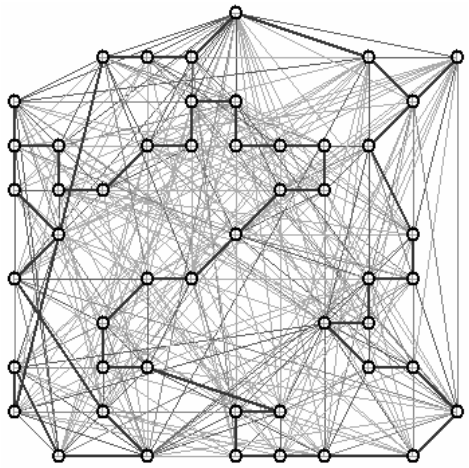


Figura 3.6: Sociograma formado pelas interações dos agentes com algoritmos de colônia de formigas

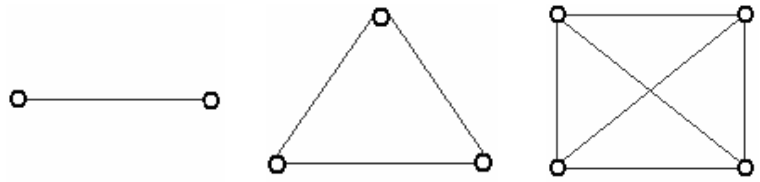
Baseado na nomenclatura utilizada em Wasserman e Faust (1994), os estados (nós) de uma rede possuem a notação e, e E é o conjunto de estados. As ligações (arestas) de uma rede têm a notação c, e o conjunto de ligações será C. Assim, uma rede de n estados e de m ligações terá um conjunto de estado E= $\{e_1,e_2,...,e_n\}$ e um conjunto de ligações C= $\{c_1,c_2,...,c_m\}$.

Quando há ligação entre dois estados, então é formado um par de estados (ou díade). Por exemplo, uma ligação c_1 pode ser referente à ligação entre os estados e_1 e e_2, que pode ser denotada por c_1= (e_1,e_2). A ligação de dois estados pode ser direcionada ou não-direcionada. Por exemplo, se a ligação c_1 é direcionada do estado e_2 para o estado e_5, será então c_1= $(e_2 \rightarrow e_5)$. Para encontrar a quantidade máxima c_{max} de ligações em um grafo simples não-direcionado, a equação 3.3 é empregada.

$$c_{max} = \frac{n(n-1)}{2} \tag{3.3}$$

[2] No problema do caixeiro viajante um estado representa uma cidade.

Dessa forma, c_{max}= 1 indica dois estados conectados; c_{max}= 3 indica três estados conectados; c_{max}= 6 quando há quatro estados conectados, e assim por diante. A figura 4.7 mostra a quantidade máxima de ligações em grafos não-direcionados.



A b c

Figura 3.7: Quantidade máxima de ligações em grafos não-direcionados

Em grafos direcionados simples, a quantidade máxima de ligações entre dois estados é de duas ligações opostas, para três estados o máximo é de seis, e assim por diante. A expressão $c_{max.dir}= n(n–1)$ define a quantidade máxima de ligações direcionadas no grafo (Knoke e Yang, 2008).

Dessa forma, os grafos fornecem métodos interessantes para a análise de redes sociais, e observações visuais podem auxiliar na compreensão da rede. No entanto, isso se torna impraticável em redes com muitos estados e ligações, e informações importantes como a intensidade da ligação e valores específicos (*e.g.*, recompensas e/ou feromônios gerados por algoritmos de otimização) são dificilmente aplicáveis no grafo. Para resolver tal problema, é possível empregar matrizes desenvolvidas pela sociometria, chamadas então de sociomatrizes ou matriz de adjacência, na qual a representação na matriz irá indicar se estados estão ou não adjacentes. Dessa maneira, a sociometria com as sociomatrizes fazem parte da teoria dos grafos, fornecendo base matemática para a análise das redes sociais (Wasserman e Faust, 1994).

Uma matriz pode mostrar as ligações entre os estados da rede. Cada elemento da matriz pode indicar a ligação entre dois estados. Cada elemento da matriz indica o valor para uma determinada linha e coluna (figura 3.9). Considerando os valores de *i* e *j*, cada elemento pode ser identificado, onde x_{ij}= 1 indica ligação entre n_i e n_j, e x_{ij}= 0 quando não há ligação. Se $x_{ij} = x_{ji}$ então a matriz é simétrica.

O uso de grafos é necessário para criar modelos ou sistemas de representação dos estados da rede. Muitas vezes, não é possível representar a totalidade das características dos atributos da rede. Para isso, conceitos adicionais podem ser usados para auxiliar na análise dos relacionamentos.

3.2.2.1 Grau do Estado

Em uma rede não-direcionada, pode-se medir o número de ligações incidentes em um estado, denominado grau do estado. O grau do estado é zero quando não há ligação com os demais estados, ou valor $n - 1$ quando existe ligação do estado com os demais. A medida do grau de um estado pode definir sua importância, como sua influência na rede.

Conhecendo a quantidade de ligações ao estado, é possível obter o grau de um determinado estado $g(e_n)$. Na figura 3.8 é possível computar a quantidade de ligações incidentes em cada estado:

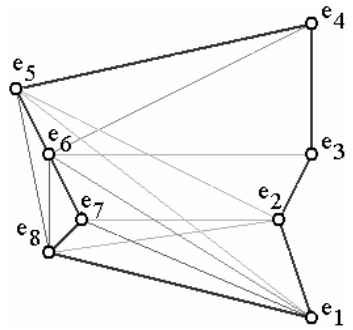


Figura 3.8: Rede não direcionada com 8 estados

no qual, $g(e_1)= 5$, $g(e_2)= 5$, $g(e_3)= 3$, $g(e_4)= 3$, $g(e_5)= 5$, $g(e_6)= 6$, $g(e_7)= 4$, $g(e_8)= 5$. A figura 3.9 apresenta a sociomatriz usada para mostrar a ligação dos estados, onde 1 indica relação. Neste trabalho, as sociomatrizes serão usadas para mostrar a intensidade da relação entre os estados.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	1	0	0	1	1	1	1
2	1	0	1	0	1	0	1	1
3	0	1	0	1	0	1	0	0
4	0	0	1	0	1	1	0	0
5	1	1	0	1	0	1	0	1
6	1	0	1	1	1	0	1	1
7	1	1	0	0	0	1	0	1
8	1	1	0	0	1	1	1	0

Figura 3.9: Sociomatriz do grafo da figura 3.8

Em grafos direcionados é possível identificar a direção da ligação. Dessa maneira é possível quantificar as ligações que chegam e que saem dos estados (Wasserman e Faust, 1994).

3.2.2.2 Densidade da rede

A densidade da rede indica a quantidade de ligações existentes, redes com alta densidade possuem grande quantidade de ligações e redes com poucas ligações são chamadas de esparsas (Wasserman e Faust, 1994). Para medir a densidade de uma rede não-direcionada, define-se a quantidade de ligações C da rede, dividida pela quantidade máxima C_{max} de ligações. A equação 3.4 é usada para computar a densidade da rede.

$$D = \frac{C}{n(n-1)/2} = \frac{2C}{n(n-1)} \tag{3.4}$$

Se a rede possui máxima ligação, a rede é então completa e o valor da densidade será 1. Se a rede não possui ligações a densidade é 0. Para uma rede direcionada a medida da densidade é definida pela quantidade de ligações da rede, dividido pela quantidade máxima. A equação 3.5 é simplificada para computar a densidade de grafos direcionados.

$$D = \frac{C}{n(n-1)} \tag{3.5}$$

3.2.2.3 Geodésico

O caminho mais curto entre dois estados é chamado de geodésico, onde o comprimento do caminho é denominado de distância geodésica (Wasserman e Faust, 1994). Tal distância permite verificar a quantidade de ligações e estados que estão intermediários aos estados. Considerando a distância geodésica entre dois estados quaisquer e_i e e_j, é possível calcular a distância d(e_i, e_j). Considerando o estado e_1 da figura 3.10, é possível computar as distâncias geodésicas aos demais estados:

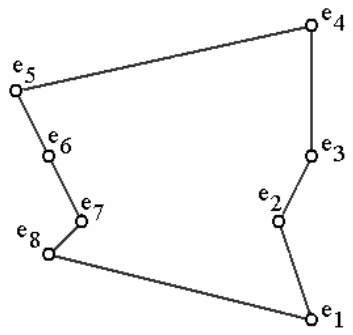


Figura 3.10: Grafo para exemplificar a distância geodésica

na qual, d(e_1, e_2)= 1; d(e_1, e_3)= 2; d(e_1, e_4)= 3; d(e_1, e_5)= 4; d(e_1, e_6)= 3; d(e_1, e_7)= 2; e d(e_1, e_8)= 1. A maior distância irá determinar o diâmetro da rede (nesse caso d(e_1, e_5)) e quando não houver caminho entre dois estados a distância é considerada infinita.

3.2.2.4 *Cutpoint* e Pontes

Um estado *cutpoint* é aquele que se excluído da rede fará com que alguns estados sejam desconectados (Wasserman e Faust, 1994). Pode haver estados *cutpoint* importantes, e se excluídos podem dividir a rede, aumentando a distância geodésica entre alguns estados. Por exemplo, considerando o estado e_4 da figura 3.10 como um *cutpoint*, a distância geodésica entre os estados e_3 e e_5, d(e_3, e_5), iria aumentar de 2 para 6.

A noção de pontes é similar à do estado de corte, no entanto, refere-se somente à exclusão de determinadas ligações, mantendo o estado na rede. Estados *cutpoint* e pontes podem ser importantes em grafos onde a busca por determinados valores são necessários, pois o seu uso poderia diminuir o espaço de busca quando muitos estados estão disponíveis. Usar *cutpoint* diminui a quantidade de ligações da rede, melhorando o processo de busca de um algoritmo.

3.2.2.5 Centralidade e Prestígio

Dois importantes conceitos em redes sociais são a centralidade e o prestígio, identificando estados importantes na rede (Wasserman e Faust, 1994). Há várias maneiras para calcular a centralidade, por exemplo, para um determinado estado e_i, a centralidade pode ser denotada como $C(e_i)$ e a medida é dada pela quantidade de ligações do estado na rede (grau do estado). A equação 3.6 é utilizada para obter a centralidade de grau:

$$C(e_i) = \frac{d(k_i)}{n-1} \tag{3.6}$$

onde n é a quantidade de estados da rede e k_i a quantidade de estados adjacentes do estado analisado. Como $C(e_i)$ é independente de n, então essa métrica pode ser usada em redes com quantidades diferentes de estados.

O conceito de prestígio de um estado e_i está atrelado às redes direcionadas, onde a direção das ligações define seu prestígio na rede. Métricas de centralidade de grau, centralidade de intermediação e centralidade de proximidade podem ser usadas em conjunto com algoritmos de colônia de formigas. Será mostrado como conceitos básicos de grafos podem ser empregados para analisar a evolução de redes sociais em problemas de otimização, tais como relacionamento, prestígio, influência e outras definições.

3.3 Abordagens Computacionais

Para facilitar a análise e a visualização das redes sociais, diversas ferramentas foram propostas, como: UCINET (Borgatti *et al.* 2002b), Pajek (Batagelj e Mrvar, 2003b), STRUCTURE (Burt, 1991), StOCNET (Huisman e Van Duijn, 2003), MultiNet (Richards e Seary, 2003) e GRADAP (Stokman e Sprenger, 1989). Há ainda aplicações mais específicas, como Netdraw (Borgatti, 2002a), SIENA (Snijders, 2001) e KrackPlot (Krackhardt *et al.* 1994). Alguns dos principais aplicativos são sumarizados na sequência. A descrição detalhada de tais aplicativos pode ser encontrada no trabalho de (Huisman e Van Duijn, 2004).

Pajek: é um aplicativo para visualização e análise de redes, especialmente desenvolvido para lidar com grandes conjuntos de dados (Batagelj e Mrvar, 2003ba). Os principais objetivos do Pajek são: i) reduzir redes imensas em várias redes menores, onde seja possível analisá-las empregando métodos estatísticos; ii) fornecer ferramentas para visualização dos dados; e iii) disponibilizar algoritmos para análise (Batagelj e Mrvar, 2002).

GRADAP (*Graph Definition and Analysis Package*): é um aplicativo para análise e definições gráficas (Stokman e Sprenger, 1989). Foi desenvolvido com a colaboração de pesquisadores das seguintes universidades: Amsterdam, Groningen, Nijmegen, e Twente. GRADAP analisa explicitamente os dados da rede representados por grafos. Para isso, inclui uma variedade de métodos de centralidade e subgrupos coesos.

Structure: é um aplicativo que fornece sociogramas, *cliques*, equivalência estrutural, tabelas de densidade e outros (Burt, 1991). Structure suporta modelos de redes com os seguintes tipos de análise: análise estrutural, coesão (detecção de *cliques*) e equivalência (análise estrutural ou equivalência, e *blockmodeling*).

UCINET: é um dos aplicativos mais abrangentes para a análise de redes sociais e aproximação de dados, pois contém um grande número de rotinas analíticas para a rede (Borgatti *et al.* 2002b).

NetMiner II: é um aplicativo que combina análise de redes sociais e técnicas de exploração visual (Cyram, 2003). O aplicativo permite explorar os dados da rede de maneira visual e interativa, ajudando a detectar padrões e estruturas da rede.

StOCNET (*StOChastic NETworks*): é um sistema aplicativo para análise estatística avançada de redes sociais (Boer *et al.* 2003). É fornecida uma plataforma que disponibiliza os métodos estatísticos apresentados em módulos, e permite que novas rotinas sejam implementadas (Huisman e Duijn, 2003). O sistema é divido em sessões, que consistem de um processo cíclico de cinco etapas: (i) definição dos dados; (ii) transformação; (iii) seleção; (iv) modelo de especificação e análise; e v) inspeções dos resultados.

Apesar desses aplicativos oferecerem as ferramentas computacionais para a visualização gráfica da rede e permitirem o uso das equações de centralidade e subgrupos, não é trivial empregar tais aplicativos em aplicações com algoritmos de otimização. Devido a isso, foi desenvolvido no trabalho em questão um *framework* que permite a visualização de sociogramas e implementa as equações essenciais da análise de redes sociais, que foram integradas a um algoritmo de colônia de formigas.

3.4 Redes Sociais e Sistemas Multiagente

Um sistema como uma sociedade de agentes, pode ser definido como uma entidade cognitiva e social, que possui relacionamentos identificáveis e ligações entre os agentes, podendo ser coordenados por algum método de coordenação (Panzarasa e Jennings, 2001).

Geralmente, as sociedades estão organizadas de acordo com alguma estrutura, tais como redes ou hierarquias. Ao contrário das hierarquias, as redes têm sido identificadas como estruturas sociais. Redes sociais bem estabelecidas permitem a seleção de grupos dos melhores agentes para a realização de determinadas tarefas. Em sociedades complexas, noções de intensidade da relação são fundamentais para a criação de uma estrutura social. Dessa forma, redes sociais podem ser consideradas essenciais para atribuir conceitos de reputação e relação entre os agentes de um sistema multiagente.

As redes sociais apresentam várias características que favorecem os métodos de coordenação para sistemas multiagente. Por exemplo, no trabalho de (Mérida-Campos e Willmott, 2004) verificou-se que as coalizões formadas com agentes que apresentavam alto grau de in-

termediação em uma rede de agentes relacionados (*betweeness*) apresentavam valores elevados de utilidade. No trabalho de (Gaston e DesJardins, 2005) é apresentado um estudo sobre como a estrutura de uma rede social tem impacto na decisão dos indivíduos em situações específicas como: difusão de inovações, formação de opinião e formação de times. Na difusão de inovação os agentes têm dois estados [1,0], *i.e.* adota ou não adota a inovação. Um agente tem maior probabilidade de adotar a inovação em função da capacidade de processamento e da quantidade de vizinhos que adotaram a inovação. Na formação de opinião é calculado o impacto social para observar a mudança de opinião do agente, baseado na força de cada agente, distância entre os agentes, na influência externa e no ruído social, sem considerar a estrutura organizacional. O agente líder (agente com a menor distância média em relação aos demais agentes) deve convencer os agentes a adotar a sua opinião mais facilmente do que agentes mais isolados. Na formação de times a topologia da rede restringe os agentes que podem participar do mesmo time. Um time é um subgrupo conectado de agentes onde a soma de todas as competências é suficiente para a execução de uma tarefa. As tarefas são distribuídas em *broadcast* e em intervalos regulares de tempo. Os resultados mostram que redes de livre escalas apresentam resultados melhores em relação à eficiência organizacional.

Estes são apenas alguns dentre os diversos trabalhos que aplicam tais conceitos para o aprimoramento de técnicas de inteligência artificial distribuída. Dentre outros trabalhos podemos citar (Dautenhahn, 1995; Dautenhahn e Christaller, 1996; Ogden e Dautenhahn, 2001; Bowman e Hexmoor, 2005; Araújo e Lamb, 2008).

3.5 Considerações Finais

Observou-se que uma rede social é composta por um conjunto de indivíduos sociais, ou agentes e seus relacionamentos, que interagem caracterizando um sistema multiagente. Como característica, um agente não atua somente de maneira autônoma, isto é, seu comportamento individual geralmente influencia no comportamento dos demais agentes, modificando a estrutura social do sistema. Ferramentas da análise das redes sociais podem identificar o grau de sociabilidade dos agentes e da estrutura social a partir dos comportamentos, servindo como base para a construção de modelos sociais.

Critérios como características pessoais, profissionais, problemas sociais, confiança, parentesco, *etc.*, são comuns para a representação de redes sociais da sociedade humana. No entanto, em sistemas multiagente, tais conceitos não fazem sentido na sua interpretação literal, podendo ser adaptados ou redefinidos. Mostramos como os conceitos relacionados às medidas

de centralidade e intensidade das relações estabelecem relações sociais entre os agentes. Além disso, outras abordagens serão discutidas adiante para estabelecer a estrutura social de um sistema multiagente.

No capítulo 4 serão apresentados os principais conceitos sobre aprendizagem por reforço e otimização por enxames, e seus principais algoritmos. Esses princípios nos ajudam a entender como um conjunto de agentes pode se coordenar com valores (recompensas) gerados a partir de seus comportamentos individuais. Será observado que a manutenção desses comportamentos é realizada por valores de recompensas que determinam as atitudes dos agentes, podendo ser compartilhadas influenciando a geração de novos comportamentos. Neste caso as recompensas são denominadas de recompensas sociais. Acreditamos que à medida que os agentes influenciam ou alteram a estrutura social compartilhando essas recompensas, eles podem melhorar seu comportamento individual e coletivo.

Capítulo 4

Aprendizagem por Reforço e Otimização por Enxames

Foi possível observar no capítulo 3 que as redes sociais são formadas por um conjunto de estados que podem se conectar através de ligações que representam relações. Essas relações podem ser constituídas por aspectos que indicam a força ou a intensidade da relação, ou ainda a frequência de incidências no estado, mostrando sua centralidade, prestígio ou influência na rede. A computação, de certa forma, tem contribuído para a aplicação sistemática de metodologias que permitem representar, quantificar e analisar estas relações como é possível verificar nos parágrafos seguintes.

> Pavlov, um cientista russo, publicou em 1903 um artigo chamado “reflexo condicional” e a sua experiência ficou mundialmente conhecida como “o cachorro de Pavlov”. No seu trabalho, ele tocava um sino toda vez que dava comida ao seu cachorro. Com o passar do tempo o cão começou a associar o som do sino com comida, aprendendo que o som estava relacionado com comida. Seguindo em sua pesquisa Pavlov ”ensinou“ ao cão que alguns dos sinais eram bons e outros ruins, então o cão começou a evitar os “sinais ruins” e aumentou o seu interesse pelos ”sinais bons” (Pavlov, 1927).

Pavlov impôs ao seu cachorro aspectos que indicam relações através de sinais, atribuindo recompensas quando a ação é boa e punições quando a ação não é desejada. Esse contexto mostra a ligação de Pavlov com seu cachorro, onde são observados conceitos da aprendizagem por reforço que podem determinar a relação pela interação de ambos.

A aprendizagem por reforço fornece as técnicas necessárias para estabelecer e intensificar as relações entre os estados de uma rede. Algoritmos como o *Q-learning* (Watkins e Dayan, 1992), podem de maneira iterativa estabelecer relações compartilhando recompensas ou

políticas quando o modelo do sistema não está definido, aprendendo uma política de ação que indica a intensidade ou a força dos estados na rede. De maneira mais geral, são apresentados na próxima seção os fundamentos essenciais para o entendimento da aprendizagem por reforço. Esses fundamentos são rediscutidos adiante, onde é mostrado como algoritmos de aprendizagem por reforço podem gerar e compartilhar recompensas de outros agentes, melhorando a política de ação e produzindo redes de relacionamentos entre estados e ações.

4.1 Definições da Aprendizagem por Reforço

A aprendizagem por reforço é um paradigma computacional de aprendizagem em que um agente aprendiz procura maximizar uma medida de desempenho baseada nos reforços (recompensas ou punições) que recebe ao interagir com um ambiente (Ribeiro, 1999). A aprendizagem por reforço vem sendo utilizada por diversos pesquisadores (Tesauro, 1995; Crites e Barto, 1996; Kaelbling *et al.* 1996; Littman e Kaelbling, 1996; Sutton e Barto, 1998; Ribeiro, 1999; Porta e Celaya, 2005; Ribeiro *et al.* 2009b) no intuito de encontrar soluções para problemas de aprendizagem com o uso de agentes.

O agente atua no ambiente formado por um conjunto de estados e pode escolher ações dentro de um conjunto de ações possíveis, indicando o valor imediato da transição de estado resultante. A tarefa do agente consiste em aprender uma política de controle (sequência de ações) que maximiza a soma esperada destes reforços, descontando (usualmente de modo exponencial) as recompensas ou punições proporcionalmente ao seu atraso temporal (Sutton e Barto, 1998).

No problema de aprendizagem por reforço tem-se um agente, que atua em um ambiente. O agente percebe um conjunto discreto S de estados, e pode realizar um conjunto discreto A de ações. A cada instante de tempo *t*, o agente pode detectar seu estado atual *s*, e, de acordo com esse estado, escolher uma ação a ser executada, que o levará para um outro estado *s*'. Para cada par estado/ação, (*s*,*a*), há um sinal de reforço dado pelo ambiente, $R(s,a) \rightarrow \Re$, que é informado ao agente quando ele executa a ação *a* no estado *s*. O problema da aprendizagem por reforço é ilustrado na figura 4.1.

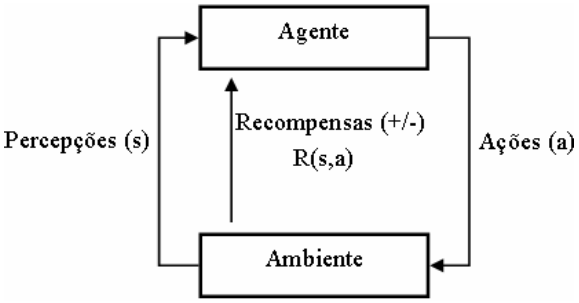


Figura 4.1: Aprendizagem por reforço (Sutton e Barto, 1998)

O sinal de reforço é a base do aprendizado do agente, pois o valor do reforço deve indicar o objetivo a ser alcançado pelo agente. O agente receberá uma recompensa positiva caso o seu novo estado seja melhor do que o seu estado anterior. Com isso, o reforço mostra ao agente que a sua meta é maximizar recompensas até o seu estado final.

Assim, o objetivo do método é levar o agente a escolher a sequência de ações que tendem a aumentar a soma de valores de reforço, ou seja, é encontrar a política ótima, definida como o mapeamento de estados em ações que maximize as recompensas acumuladas no tempo.

4.1.1 Características da Aprendizagem por Reforço

Algumas características que diferenciam a aprendizagem por reforço de outros métodos são descritas a seguir (Sutton e Barto, 1998):

- Aprendizado pela interação: essa é a principal característica que define um problema de aprendizagem por reforço, onde um agente age no ambiente e aguarda pelo valor de reforço que o ambiente deve informar como resposta pela ação tomada, assimilando através do aprendizado o valor de reforço obtido para tomar decisões posteriores;
- Retorno atrasado: um valor máximo de reforço que o ambiente envia para o agente não indica necessariamente que a ação tomada pelo agente foi a melhor. Uma ação é produto de uma decisão local no ambiente, sendo seu efeito imediato de natureza local, enquanto que em um sistema de aprendizagem por reforço, busca-se alcançar objetivos globais no ambiente. Assim as ações tomadas devem levar a maximizar o retorno total, isto é, a qualidade das ações tomadas é vista pelas soluções encontradas em longo prazo;
- Orientado a objetivo: em aprendizagem por reforço, o problema tratado é considerado como um ambiente que dá respostas frente às ações efetuadas, não sendo necessário conhecer detalhes da modelagem desse ambiente. Simplesmente, existe um agente que

age dentro do ambiente tentando alcançar um objetivo. O objetivo é, geralmente, otimizar algum comportamento; e

- Investigação x exploração[3]: em aprendizagem por reforço, agentes presenciam o dilema conhecido na literatura como "*the Exploration* x *Exploitation dilemm*", que consiste em decidir quando se deve aprender sobre o ambiente, usando informações obtidas até o momento.

A decisão é fundamentalmente uma escolha entre agir baseado na melhor informação de que o agente dispõe no momento ou agir para obter novas informações sobre o ambiente que possam permitir níveis de desempenho maiores no futuro. Isto significa que o agente deve aprender quais ações maximizam os valores dos ganhos obtidos no tempo, mas também, deve agir de forma a atingir esta maximização, explorando ações ainda não executadas ou regiões pouco visitadas do espaço de estados. Como ambas as formas trazem, em momentos específicos, benefícios à solução dos problemas, uma boa estratégia é mesclar estas formas (Sutton e Barto, 1998).

4.1.2 Elementos Fundamentais da Aprendizagem por Reforço

Conforme apresentado em Sutton e Barto (1998), o problema da aprendizagem por reforço apresenta cinco partes fundamentais a serem consideradas: (i) o ambiente, (ii) a política, (iii) o reforço e o retorno, (iv) a função de reforço e (v) a função valor-estado, descritos assim:

i) O ambiente: todo sistema de aprendizagem por reforço aprende um mapeamento de situações e ações por experimentação em um ambiente. O ambiente no qual está inserido o sistema, deve ser pelo menos parcialmente observável através de sensores ou descrições simbólicas. Também é possível, entretanto, que toda informação relevante do ambiente esteja perfeitamente disponível. Neste caso, o agente poderá escolher ações baseadas em estados reais do ambiente.

ii) A política: uma política sendo expressa pelo termo π, representa o comportamento que o sistema de aprendizagem por reforço segue para alcançar o objetivo. Em outras palavras, uma política π é um mapeamento de estados *s* e ações *a* em um valor $\pi(s,a)$. Assim, se um agente altera a sua política, então as probabilidades de seleção de ações sofrem mudanças e consequentemente, o comportamento do sistema apresenta variações à medida que o agente

[3] Exploração em algoritmos de busca é escolher caminhos não visitados e investigação (explotação) é optar pelo caminho da melhor solução.

vai acumulando experiência a partir das interações com o ambiente. Portanto, o processo de aprendizagem por reforço pode ser expresso em termos da convergência até uma política ótima $\pi^*(s,a)$ que conduz à solução do problema de forma ótima.

iii) Reforço e retorno: o reforço é um sinal do tipo escalar r_{t+1}, que é devolvido pelo ambiente ao agente assim que uma ação tenha sido efetuada e uma transição de estado $s_t \rightarrow s_{s+1}$ tenha ocorrido. Existem diferentes formas de definir o reforço para cada transição no ambiente, gerando-se funções de reforço que, intrinsecamente, expressam o objetivo que o sistema de aprendizagem por reforço deve alcançar. O agente deve maximizar a quantidade total de reforços recebidos chamado de retorno, que nem sempre significa maximizar o reforço imediato a receber, mas o reforço acumulado durante a execução total.

De modo geral, a aprendizagem por reforço busca maximizar o valor esperado de retorno, com isso, o retorno pode ser definido como uma função da sequência de valores de reforço até um tempo T final. No caso mais simples é um somatório como demonstrado na equação 4.1.

$$R = \sum_{k=0}^{T} r_{t+k+1} \tag{4.1}$$

Em muitos casos a interação entre agente e ambiente não termina naturalmente em um episódio (sequência de estados que chegam até o estado final), mas continua sem limite, como por exemplo, em tarefas de controle contínuo. Para essas tarefas a formulação do retorno é um problema, pois $T = \infty$ e o retorno que se deseja também tenderá a infinito $R_t = \infty$.

Para este problema foi criada a taxa de amortização *y*, a qual determina o grau de influência que têm os valores futuros sobre o reforço total. Assim, a expressão do retorno aplicando taxa de amortização é expressa pela equação 4.2:

$$R = \sum_{k=0}^{\infty} y^k r_{t+k+1} \tag{4.2}$$

na qual, $0 \leq y \leq 1$. Se $y \rightarrow 0$, o agente tem uma visão baixa dos reforços, maximizando apenas os reforços imediatos. Se $y \rightarrow 1$, a visão do reforço abrange todos os estados futuros dando maior importância ao estado final, desde que a sequência R seja limitada. Um sistema de

aprendizagem por reforço faz um mapeamento de estados em ações baseado nos reforços recebidos.

Assim, o objetivo da aprendizagem por reforço é definido usando-se o conceito de função dos reforços futuros que o agente procura maximizar. Ao maximizar essa função, o objetivo será alcançado de forma ótima. A função de reforço define quais são os bons e maus eventos para os agentes.

iv) Função de Reforço: as funções de reforço podem ser bastante complexas, porém existem pelo menos três classes de problemas frequentemente usados para criar funções adequadas a cada tipo de problema, descritas assim:

- Reforço só no estado final: nesta classe de funções, as recompensas são todas iguais a zero, exceto no estado final, em que o agente recebe uma recompensa real (*e.g.*, +1) ou uma penalidade (*e.g.*, –1). Como o objetivo é maximizar o reforço, o agente irá aprender que os estados correspondentes a uma recompensa são bons e os que levaram a uma penalidade devem ser evitados;
- Tempo mínimo ao objetivo: funções de reforço nesta classe fazem com que o agente realize ações que produzam o caminho ou trajetória mais curta para um estado final. Toda ação tem penalidade –1, sendo que o estado final é 0. Como o agente tenta maximizar valores de reforço, ele aprende a escolher ações que minimizam o tempo que leva para alcançar o estado final; ou
- Minimizar reforços: nem sempre o agente precisa ou deve tentar maximizar a função de reforço, podendo também aprender a minimizá-las. Isto é útil quando o reforço é uma função que representa recursos limitados e o agente deve aprender a conservá-los ao mesmo tempo em que alcança o estado final.

v) Função Valor-Estado: define-se uma função valor-estado como o mapeamento do estado, ou par estado-ação, em um valor que é obtido a partir do reforço atual e dos reforços futuros. Se a função valor-estado considera só o estado s é indicada como $V(s)$, por outro lado, se é considerado o par estado-ação (s,a), então a função valor-estado é denotada como função valor-ação $Q(s,a)$.

4.1.3 Processos Markovianos

A maneira mais tradicional para formalizar a aprendizagem por reforço consiste em utilizar o conceito de processos decisórios de *Markov*. Por ser matematicamente bem estabelecido e fundamentado, este formalismo facilita o estudo da aprendizagem por reforço. Por ou-

tro lado, assume uma condição simplificadora, conhecida como condição de *Markov*, que reduz a abrangência das soluções, mas que é compensada em grande parte pela facilidade de análise (Ribeiro, 2002).

A condição de *Markov* especifica que o estado de um sistema no próximo instante (t+1) é uma função que depende somente do que se pode observar acerca do estado atual e da ação tomada pelo agente neste estado (descontando alguma perturbação aleatória), isto é, o estado do sistema independe de sua história. Pode-se ver que muitos domínios obedecem esta condição: problemas de roteamento, controle de inventário, escalonamento, robótica e problemas de controle discreto em geral.

Um processo decisório de *Markov* é aquele que obedece à condição de *Markov* e pode ser descrito como um processo estocástico no qual a distribuição futura de uma variável depende somente do seu estado atual (Littman, 1994; Mitchell, 1997). Um processo decisório de *Markov* é definido formalmente pela quádrupla < S,A,T,R >, onde:

- S: é um conjunto finito de estados do ambiente;
- A: é um conjunto finito de ações que o agente pode realizar;
- T:SxA$\rightarrow \Pi$(S): é a função de transição de estado, onde Π(S) é uma distribuição de probabilidades sobre o conjunto de estados S e $T(s_{t+1}, s_t \mid a_t)$ define a probabilidade de realizar a transição do estado s_t para o estado s_{t+1} quando se executa a ação a_t; e
- R:SxA$\rightarrow \Re$: é a função de recompensas, que especifica a tarefa do agente, definindo a recompensa recebida por um agente ao selecionar a ação *a* estando no estado *s*.

Usando o processo decisório de *Markov* como formalismo, pode-se definir a capacidade do agente que aprende por reforço como: capacidade de aprender a política π^*: SxA que mapeia o estado atual s_t em uma ação desejada, de forma a maximizar a recompensa acumulada ao longo do tempo, descrevendo o comportamento do agente (Kaelbling *et al.* 1996).

Um processo decisório de *Markov* pode ser determinístico ou não-determinístico, dependendo da função de probabilidade de transição $T(\cdot)$. Caso $T(\cdot)$ especifique apenas uma transição válida para um par (estado-ação), o sistema é determinístico; caso a função defina um conjunto de estados sucessores potencialmente resultantes da aplicação de uma determinada ação em um estado, o sistema é chamado de não-determinístico. Um exemplo deste último pode ser dado para o domínio do futebol de robôs, no qual uma bola chutada em direção ao gol pode entrar, pode bater no travessão ou pode ir para fora do campo. Outro exemplo é do lançamento de uma moeda, no qual dois resultados são possíveis.

4.2 Algoritmos de Aprendizagem por Reforço

A aprendizagem por reforço dispõe de vários algoritmos de aprendizagem, como *Q-learning* (Watkins e Dayan, 1992), *R-learning* (Schwartz, 1993), *H-learning* (Tadepalli e Ok, 1994), *Sarsa* (Sutton e Barto, 1998), *Dyna* (Singh e Sutton, 1996) entre outros. Na sequência, esses algoritmos são sumarizamos e descrito o *Q-learning* devido sua similaridade e inspiração para os algoritmos de colônia de formiga.

4.2.1 Algoritmo *Q-learning*

O algoritmo *Q-learning* proposto por Watkins e Dayan em 1992 é o método mais popular utilizado para problemas de aprendizagem por reforço (Watkins e Dayan, 1992). Trata-se de um algoritmo que permite estabelecer de maneira autônoma e iterativa uma política de ações. Pode-se demonstrar que o algoritmo *Q-learning* converge para um procedimento de controle ótimo, quando a hipótese de aprendizagem de pares estado-ação Q for representada por uma tabela completa contendo a informação de valor de cada par. A convergência ocorre tanto em processos de decisão *Markovianos* determinísticos quanto não-determinísticos.

A ideia básica do *Q-learning* é que o algoritmo de aprendizagem aprenda uma função de avaliação ótima sobre todo o espaço de pares estado-ação SxA. A função Q fornece um mapeamento da forma Q:SxA→V, onde V é o valor de utilidade esperada ao se executar uma ação a no estado s. Desde que o particionamento do espaço de estados do agente e o particionamento do espaço de ações não omitam informações relevantes, uma vez que a função ótima seja aprendida, o agente saberá que ação resultará na maior recompensa futura em uma situação particular s.

A função Q(s,a), da recompensa futura esperada ao se escolher a ação a no estado s, é aprendida através de tentativa e erro, conforme equação 4.3:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + y.\max_a Q(s',a) - Q(s,a)] \qquad (4.3)$$

no qual α é a taxa de aprendizagem, r é a recompensa, ou custo, resultante de tomar a ação a no estado s, y é o fator de desconto e o termo $\max_a Q(s',a)$ é a utilidade do estado s resultante da ação a, obtida utilizando a função Q que foi aprendida até o presente. A função Q representa a recompensa descontada esperada ao se tomar uma ação a quando visitando o estado s, e seguindo-se uma política ótima desde então.

O fator *y* pode ser interpretado de várias formas: pode ser visto como uma taxa de gratificação, como uma probabilidade de ir para o próximo estado ou como um artifício matemático para evitar a soma infinita (Watkins e Dayan, 1992). A forma procedimental do algoritmo *Q-learning* é apresentada na figura 4.2.

```
Algoritmo Q-learning()
01  Para cada s,a inicialize Q(s,a)=0
02  Percebe s
03  Repita até que critério de parada seja satisfeito:
04    Selecione ação a usando a política de ações atual
05    Execute a ação a
06    Receba a recompensa imediata r(s,a)
07    Observe o novo estado s'
08    Atualize Q(s,a) de acordo com a equação 4.3
09    s ← s'
10   Se critério de parada falso retorne ao passo 3
11  Fim
```

Figura 4.2: Algoritmo *Q-learning* (adaptado de Watkins e Dayan (1992))

Uma vez que todos os pares estado-ação tenham sido visitados um número finito de vezes, garante-se que o método gerará estimativas Q_t que convergem para o valor de Q^* (Watkins e Dayan, 1992). Na prática, a política de ações converge para a política ótima em tempo finito, embora de forma lenta.

Uma característica do *Q-learning* é a função valor-ação *Q* aprendida, que se aproxima diretamente da função valor-ação ótima Q^*, sem depender da política que está sendo utilizada. Este fato simplifica bastante a análise do algoritmo e permite fazer testes iniciais da convergência. A política ainda mantém um efeito ao determinar quais pares estado-ação devem ser visitados e atualizados, porém, para que a convergência seja garantida, é necessário que todos os pares estado-ação sejam visitados continuamente e atualizados.

Dados os valores *Q*, existe uma política definida pela execução da ação a, quando o agente está em um estado *s*, que maximiza o valor Q(*s*,*a*). Watkins e Dayan (1992) demonstraram que se cada par estado-ação for visitado um número suficientemente grande de vezes e *a* decrescer apropriadamente, as funções de valoração de *Q* irão convergir com certa probabilidade para Q^* e, consequentemente, a política irá convergir para uma política ótima.

A convergência do algoritmo *Q-learning* não depende somente do método de exploração usado. Um agente pode explorar suas ações a qualquer momento, não existindo requisitos para a execução de ações estimadas como as melhores. No entanto, para melhorar o desempe-

nho do sistema é necessária, durante o aprendizado, a busca das ações que maximizam o retorno.

Resumidamente, podem-se enumerar alguns dos aspectos mais importantes do algoritmo *Q-learning*:

- O objetivo do uso do algoritmo *Q-learning* é achar uma regra de controle que maximize cada ciclo de controle;
- O uso do reforço imediato é indicado sempre que possível e necessário, desde que ele contenha informação suficiente que auxilie o algoritmo a encontrar a melhor solução;
- O algoritmo *Q-learning* é adotado quando o número de estados e ações a serem selecionados é finito e pequeno;
- O algoritmo *Q-learning* foi o primeiro método de aprendizagem por reforço a possuir provas de convergência. É uma técnica simples que calcula diretamente as ações sem o uso de modelo.

4.2.2 Algoritmo *R-learning*

A técnica proposta por (Schwartz, 1993), chamada de *R-learning*, maximiza a recompensa média a cada passo, ou seja, utiliza o modelo de recompensa média. O algoritmo *R-learning* possui regra similar ao *Q-learning*, sendo baseado na dedução de valores R(*s*,*a*), e devendo escolher uma ação *a* em um estado *s*. A cada situação, o agente escolhe a ação que tem o maior valor R, exceto em algumas vezes quando ele escolhe uma ação qualquer. Os valores de R são ajustados a cada ação, baseado na seguinte regra de aprendizagem, conforme indica a equação 4.4.

$$R(s,a) \leftarrow (1-\alpha)R(s,a) + \alpha[r - \rho + eR(s')] \tag{4.4}$$

Esta regra difere da regra do *Q-learning*, simplesmente por subtrair a recompensa média ρ do reforço imediato r e por não ter desconto y para o próximo estado, $eR(s') = \max_a R(s',a')$. A recompensa média é calculada como:

$$\rho \leftarrow (1-\beta)\rho + \beta[r + eR(s') - eR(s)] \tag{4.5}$$

O ponto chave da equação 4.5 é que ρ somente é atualizado quando uma ação não aleatória foi tomada, ou seja, $\max_a R(s,a) = R(s,a)$. A recompensa média ρ não depende de um estado particular, ela é uma constante para todo o conjunto de estados. A figura 4.3 apresenta o algoritmo *R-learning*, no qual se podem observar pequenas reestruturações nas equações de atualização de R e ρ, que melhoram o custo computacional.

Algoritmo *R-Learning*

01 Inicialize ρ e R(s,a) arbitrariamente
02 Repita até condição de parada ser alcançada:
03 $s \leftarrow$ estado atual
04 Escolha $a \in A(s)$
05 Execute a ação a
06 Observe os valores s' e r
07 $R(s,a) \leftarrow R(s,a) + \alpha[r - \rho + \max_{a'} R(s',a') - R(s,a)]$
08 se $R(s,a) = \max_a R(s,a)$ então
09 $\rho \leftarrow \rho + \beta[r - \rho + \max_{a'} R(s',a') - \max_a R(s,a)]$
10 Fim

Figura 4.3: Algoritmo *R-learning* (Schwartz, 1993)

4.2.3 Algoritmo *H-learning*

O algoritmo *H-learning* foi proposto em (Tadepalli e Ok, 1994) na tentativa de otimizar a recompensa média sem utilizar descontos. O algoritmo *H-learning* estima as probabilidades $P(s'|s,a)$ e os reforços R(s,a) por contagem direta e atualiza os valores da recompensa esperada h utilizando a equação 4.6, que segundo teorema demonstrado em (Bertsekas, 1987), converge para uma política ótima.

$$h(s) = \max_{u \in A(s)} \{r(s,a) + \sum_{s'=1}^{n} P(s'|s,a)h(s')\} - \rho \qquad (4.6)$$

O algoritmo *H-learning* pode ser observado na figura 4.4:

Algoritmo *H-learning*

01 Se a estratégia de exploração sugere uma ação aleatória então
02 Selecione uma ação aleatória para i
03 Senão execute a ação $a \in_{\max} (i)$
04 Faça k ser o estado resultante e r' a recompensa imediata recebida:
05 $N(i,a) \leftarrow N(i,a)+1$
06 $N(i,a,k) \leftarrow N(i,a,k)+1$
07 $P_{ik}(a) \leftarrow N(i,a,k) / N(i,a)$
08 $r(i,a) \leftarrow r(i,a) + (r' - r(i,a)) / N(i,a)$
09 Se a ação executada $a \in_{\max} (i)$ Então
10 $T \leftarrow T+1$
11 $\rho \leftarrow \rho + (r' - h(i) + h(k) - \rho) / T$
12 Faça $H(i,u) = r(i,u) + \sum_{j=1}^{n} p_{ij}(u) h(j)$
13 $U_{\max}(i) \leftarrow \{v \mid H(i,v) = \max_{u \in U(i)} H(i,u)\}$
14 $h(i) \leftarrow H(i,a) - \rho$, onde $a \in_{\max} (i)$
15 $i \leftarrow k$
16 Fim

Figura 4.4: Algoritmo *H-learning* (Tadepalli e Ok, 1994)

Neste algoritmo N(*i*,*u*) é o número de vezes que a ação *u* foi executada no estado *i* e N(*i*,*u*,*j*) o número de vezes que ela resultou no estado *j*, $p_{ij}(u)$ é a probabilidade de ir de um estado *i* para um estado *j* executando a ação *u*, r(*i*,*a*) é a recompensa estimada por executar a ação *a* no estado *i*, h(*i*) é a recompensa máxima esperada para o estado *i* e corresponde ao *e*Q(*s*') no algoritmo *Q-learning*, T é o número total de passos que uma ação aparentemente ótima foi executada e é inicializada com zero.

Todos os métodos em AR, exceto o *H-learning*, tem um ou mais parâmetros, como por exemplo, o *Q-learning* tem α e *y* e o *R-learning* tem α e β. O desempenho desses algoritmos é sensível a estes parâmetros, e consequentemente é necessário ajustá-los para obter um melhor desempenho.

4.2.4 Algoritmo $Q(\lambda)$

O algoritmo $Q(\lambda)$ proposto em (Peng e Williams, 1996), é caracterizado por ser uma adaptação de uso de traços de elegibilidade para o algoritmo *Q-learning*. Traços de elegibilidade são registros temporários da ocorrência de um evento, como por exemplo, visita a um estado ou a execução de uma ação. O traço marca os parâmetros de memória associados aos

eventos como estados elegíveis para passar por mudanças no aprendizado. Quando um passo de aprendizado ocorre, apenas os estados ou ações elegíveis recebem o crédito pela recompensa ou a culpa pelo erro.

Do ponto de vista teórico, traços de elegibilidade são como uma ponte entre os métodos de Monte Carlo (Rubinstein, 1981) e de diferenças temporais (Sutton e Barto, 1998), onde se enquadram os algoritmos *Q-learning* e o *Sarsa*. Quando métodos de diferenças temporais são incrementados com traços de elegibilidade, eles produzem uma família de métodos atravessando um espectro que têm métodos de Monte Carlo em uma ponta e métodos de diferenças temporais na outra (Ribeiro, 1999). Neste intervalo estão métodos que herdam vantagens de ambos os extremos, frequentemente apresentando melhor desempenho.

Métodos de Monte Carlo podem apresentar vantagens para lidar com processos não-*Markovianos*, porque não atualizam estimativas baseados em valores estimados anteriormente. A principal desvantagem destes métodos é o grande esforço computacional. Métodos que usam traços de elegibilidade buscam, portanto, combinar a vantagem da rapidez relativa de aprendizado dos métodos de diferenças temporais e a capacidade de lidar com reforços atrasados ou observabilidade parcial dos métodos Monte Carlo (Monteiro e Ribeiro, 2004).

4.2.5 Algoritmo *Sarsa*

O algoritmo *Sarsa* é uma modificação do algoritmo *Q-learning* que utiliza um mecanismo de iteração de política (Sutton e Barto, 1998). A função de atualização do algoritmo *Sarsa* obedece a equação 4.7.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_t + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)] \qquad (4.7)$$

A forma procedimental do algoritmo *Sarsa* é similar a do algoritmo *Q-learning*. Idealmente, o algoritmo *Sarsa* converge para uma política e valor de função de ação ótima. Assim que todos os pares estado-ação tenham sido visitados um número finito de vezes e a política de escolha da próxima ação convirja, no limite, para uma política que utiliza a melhor ação (ou seja, aquela que maximiza a recompensa futura esperada).

Naturalmente, caso a ação escolhida a_{t+1} seja max $a_{t+1}Q(s_{t+1}, a_{t+1})$, este algoritmo será equivalente ao do *Q-learning* padrão. Entretanto, o algoritmo *Sarsa* admite que a_{t+1} seja escolhido aleatoriamente com uma probabilidade predefinida. Por eliminar o uso do operador max

sobre as ações, este método pode ser mais rápido que o *Q-learning* para situações onde o conjunto de ações tenha cardinalidade alta.

4.2.6 Algoritmo *Dyna*

O termo *Dyna* foi introduzido em (Singh e Sutton, 1996), e define uma técnica simples para integrar funções de aprendizado, planejamento e atuação. O agente interage com o ambiente gerando experiências. Estas experiências são utilizadas para melhorar diretamente as funções de valor e política de ações (através de algum método de AR) e aperfeiçoar um modelo do ambiente, que o agente pode usar para prever como o ambiente responderá a suas ações. As experiências originárias de simulação sobre este modelo são então utilizadas para melhorar as funções de valor e política de ações (planejamento sobre o modelo).

Após cada transição $s_t,a_t \rightarrow s_{t+1},r_t$, o algoritmo *Dyna* armazena em uma tabela, para o valor de (s_t,a_t), a transição observada (s_{t+1},r_t). Durante o planejamento, o algoritmo escolhe amostras aleatórias de pares estado-ação que foram experimentados anteriormente, ou seja, contidos no modelo. A seguir, realiza experiências simuladas nestes pares estado-ação selecionados. Finalmente, é aplicada uma atualização baseada em um método de aprendizagem por reforço sobre essas experiências simuladas. Tipicamente, o mesmo método de aprendizagem por reforço é utilizado tanto para o aprendizado a partir da experiência gerada quanto para o planejamento das experiências simuladas (Monteiro e Ribeiro, 2004).

Técnicas e funções de cálculo de recompensas também têm sido empregadas em outras abordagens com agentes, principalmente como forma de aprimorar modelos de coordenação. As seções 4.3 e 4.4 apresentam o modelo de coordenação baseado em enxames, em especial algoritmos de colônia de formigas, onde funções de recompensas auxiliam a reprodução de estratégias de exploração e comportamento emergente.

4.3 Inteligência Baseada em Enxames

A Inteligência de Enxames é um modelo de resolução de problemas baseado no comportamento coletivo e social de agentes reativos inseridos em ambientes dinâmicos (Kennedy e Eberhart, 2001). A inteligência de enxames é inspirada na natureza, onde grupos de animais como bando de pássaros, cardume de peixes e colônia de formigas conseguem sobreviver através de interações de grupo, e deste modo alcançar um determinado objetivo global. Este grupo de agentes é denominado de *Swarm* (Beni e Wang, 1989).

Agentes de enxames podem comunicar-se com seus pares (diretamente ou indiretamente), agindo sobre o ambiente que estão inseridos. Esses agentes seguem regras simples, e embora não exista uma estrutura de controle centralizado, possuem enorme capacidade de auto-organização, o que torna esse método robusto e desejável para problemas computacionais (Kennedy e Eberhart, 2001).

O objetivo dos modelos computacionais baseados em enxames é modelar o comportamento dos indivíduos, e de suas interações locais com o ambiente e com seus vizinhos mais próximos. Desta forma, o comportamento de grupo é desejável para que possa ser utilizado na busca por soluções de problemas complexos.

Técnicas baseadas em inteligência de enxames podem ser utilizadas em diversas aplicações em problemas de otimização e busca. Por exemplo, o método de otimização por enxames de partículas é modelado por dois comportamentos (Engelbrecht, 2005): i) cada indivíduo se aproxima do vizinho mais próximo que tenha o melhor conhecimento sobre o ambiente; e ii) caso o estado atual não apresente melhores resultados, o indivíduo retorna para o estado anterior. Como resultado, o comportamento coletivo que emerge é aquele no qual todos os indivíduos agem de maneira coerente, ou seja, o melhor comportamento para todos os indivíduos.

Um exemplo dessa abordagem é a otimização por colônia de formigas, que modela o comportamento de formigas que seguem o caminho com a maior concentração de feromônio, agindo localmente por probabilidade. Deste modo, o comportamento dos agentes emerge de forma a encontrar a melhor alternativa (caminho de menor custo) dentre as soluções candidatas (Dorigo, 1992).

A seguir, discutimos algumas das principais abordagens de inteligência de enxames.

4.3.1 Otimização por Enxames de Partículas

A otimização por enxames de partículas é um método de coordenação inspirado no comportamento e na dinâmica dos movimentos dos pássaros, insetos e peixes, composto por diversos indivíduos presentes em ambientes desconhecidos e altamente dinâmicos. A otimização por enxames de partículas proposta por Kennedy e Eberhart em 1995, foi originalmente desenvolvida para problemas de otimização com variáveis contínuas (Kennedy e Eberhart, 1995; Eberhart e Kennedy, 1995).

A formação de grupos tem sido observada em muitas espécies de animais. Algumas espécies, times ou grupos, são controlados por um líder, tais como grupos de leões, bando de macacos, entre outros. Nessas sociedades o comportamento dos indivíduos é fortemente base-

ado em hierarquias (Coello *et al.* 2004). No entanto, é interessante observar o comportamento de auto-organização de espécies que vivem em grupos onde o líder não é identificado, como por exemplo, bandos de pássaros, cardumes de peixes e rebanho de ovelhas. Tais grupos sociais de indivíduos não têm conhecimentos do comportamento global do grupo, e também não possuem informação global do ambiente.

Um grande número de estudos de comportamentos coletivos sociais tem sido realizado, dentre eles destacam-se bando de pássaros migratórios e cardume de peixes.

4.3.2 Inteligência Baseada em Cardume de Peixes

Um dos modelos de enxames desenvolvidos recentemente é baseado no *Fish School Search* (FSS), inspirado em cardume de peixes para realizar buscas no espaço de estados (Bastos Filho *et al.* 2008). O algoritmo utilizado no FSS é baseado em agentes reativos que se movem pelo espaço de estados assim como em outros métodos baseados na natureza.

Como em situações reais, os peixes do FSS são atraídos pela comida que é colocada no aquário em concentrações diferentes. A fim de encontrar grandes quantidades de comida, os peixes do cardume realizam movimentos independentes. Como resultado, cada peixe pode crescer ou diminuir em peso, dependendo de seu sucesso ou falha na busca por comida.

Bastos Filho *et al.* (2008) propõem a equação 4.8 para representar o ganho e a perda de peso dos peixes ao longo do tempo:

$$W_i(t+1) = W_i(t) + \frac{f[x_i(t+1)] - f[x_i(t)]}{\max\{|\, f[x_i(t+1)] - f[x_i(t)]\,|\,\}} \tag{4.8}$$

onde $W_i(t)$ representa o peso do peixe i no tempo t, $x_i(t)$ é a posição do peixe i e $f[x_i(t)]$ avalia a função de aptidão (quantidade de comida) em $x_i(t)$.

Algumas medidas adicionais foram inclusas para assegurar a convergência sobre áreas interessantes do aquário, as quais fazem com que haja uma variação no peso do peixe a cada ciclo do FSS. Um parâmetro adicional, nomeado peso escalar (W_{scale}) foi criado para limitar o peso de um peixe. O peso do peixe pode variar entre 1 e W_{scale}. Todos os peixes nascem com peso igual a $\frac{W_{scale}}{2}$.

O instinto natural dos animais é reagir a estímulos (ou algumas vezes, à falta dele). No FSS a movimentação realizada pelos peixes é considerada uma forma de reagir em relação à sobrevivência, tais como alimentação, reprodução, fuga de predadores, movimentação para

regiões habitáveis, entre outras. Movimentos individuais ocorrem para cada peixe do aquário a cada ciclo do algoritmo FSS. A direção que os peixes nadam é escolhida aleatoriamente. Depois que todos os peixes se moveram individualmente, uma média de seus movimentos é realizada, baseada no sucesso instantâneo de todos os peixes do cardume. Quando a média é calculada, uma nova direção é computada, e então cada peixe é reposicionado. Este movimento é calculado na equação 4.9:

$$x_i(t+1) = x_i(t) + \frac{\sum_{i=1}^{N} \Delta x_{ind\,i}\{f[x_i(t+1)] - f[x_i(t)]\}}{\sum_{i=1}^{N}\{f[x_i(t+1)] - f[x_i(t)]\}} \tag{4.9}$$

onde $\Delta x_{ind\,i}$ representa o deslocamento do peixe *i* devido ao movimento individual a cada ciclo do FSS.

A reprodução no FSS, como na natureza, pode ser vista como um forte indicador de que "as coisas vão bem". No FSS a procriação ocorre entre o par de peixes que se encontra em um determinado ponto do espaço de busca. A prole que passa atuar no ambiente herda os conhecimentos de seus pais. O tamanho do novo peixe *k* é dado através da média do tamanho de seus pais *i* e *j*, computado pela equação 4.10. Esse novo indivíduo é posicionado entre seus pais, onde tal posição é calculada pela equação 4.11.

$$W_k(t+1) = \frac{W_i(t) + W_j(t)}{2} \tag{4.10}$$

e,

$$x_k(t+1) = \frac{x_i(t) + x_j(t)}{2} \tag{4.11}$$

A fim de manter o número de peixes constante no cardume, à medida que um novo peixe nasce o menor peixe do aquário é removido.

4.3.3 Otimização por Colônia de Formigas

A otimização por colônia de formigas (Dorigo *et al.* 1991a; Dorigo, 1992; Dorigo *et al.* 1996; Dorigo *et al.* 1999) é uma abordagem baseada em população aplicada em vários problemas de otimização combinatória. Em outras palavras, a otimização por colônia de formigas

é uma metaheurística[4] para a solução de problemas combinatórios, inspirada no comportamento de um grupo de formigas na busca por alimento (objetivo).

Estudos sobre o comportamento forrageiro entre várias espécies de formigas mostram que elas seguem um padrão de decisão baseado na aleatoriedade (Dorigo, 1992), à medida que uma fonte de alimento é localizada, as formigas utilizam um mecanismo indireto de comunicação, denominado feromônio, que induz as formigas a seguirem o caminho indicado. Esse comportamento emergente é resultado de um mecanismo de recrutamento, onde formigas influenciam outras a seguirem em direção às fontes de alimento pelo caminho mais curto (Gambardella *et al.* 1997a).

Quando formigas localizam uma fonte de alimento, elas carregam a comida até o ninho e vão depositando o feromônio. Dessa forma, as formigas irão seguir o caminho baseado na concentração de feromônio no ambiente. Portanto, quanto maior a quantidade de formigas seguindo o mesmo caminho, maior a probabilidade do caminho ser escolhido, aumentando a qualidade e atraindo mais formigas.

A figura 4.5 ilustra a experiência realizada por (Goss *et al.* 1989) para estudar o comportamento das formigas. Inicialmente, as formigas exploram aleatoriamente a área ao redor do formigueiro à procura de comida. Enquanto se deslocam, depositam sobre o ambiente uma quantidade de feromônio, que indica a direção de retorno ao formigueiro. Desta forma, quando uma formiga estabelece um caminho entre a fonte de alimento e o formigueiro, o caminho percorrido fica indicado por rastros de feromônios. As demais formigas podem detectar a presença do feromônio no ambiente e assim tendem a escolher esse caminho.

Portanto, as formigas que escolheram o caminho mais curto farão o percurso em menor tempo, e o rastro de feromônio será aumentado com mais frequência. Por ser uma substância que evapora ao longo do tempo, caminhos que não são mais utilizados deixam de influenciar na decisão das formigas.

[4] As metaheurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo de uma heurística que deve ser modelada para cada problema específico (Dorigo, 1992).

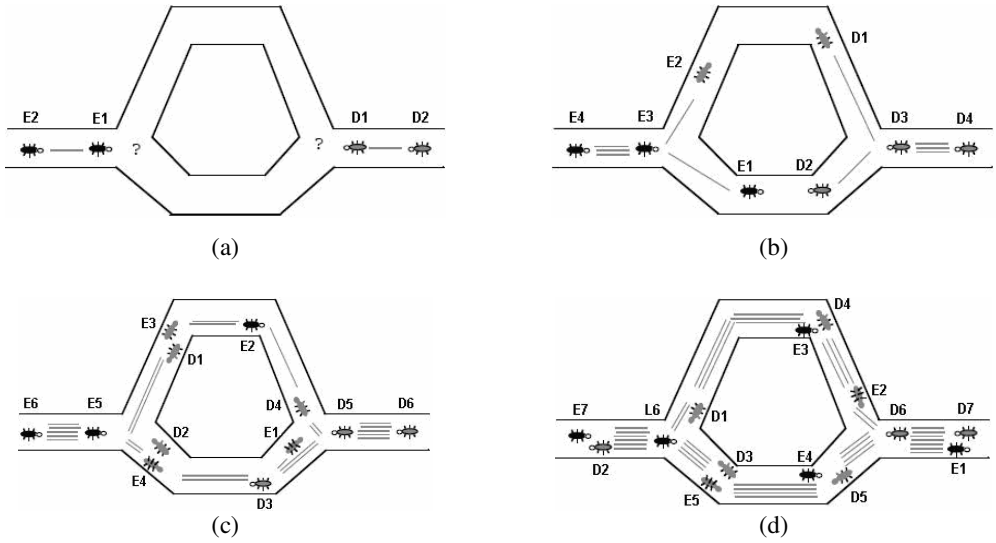


Figura 4.5: Comportamento de formigas reais (Goss *et al.* 1989)

A figura 4.5 mostra como formigas reais encontram o caminho mais curto. As formigas se movem da direita para esquerda (D) e da esquerda para direita (E). Na figura 4.5 (a) as formigas localizam no ambiente um local com alternativas diferentes para alcançar o formigueiro. Na figura 4.5 (b) as formigas escolhem de maneira aleatória o caminho a seguir. As formigas que escolheram o caminho mais curto (menor custo) alcançam o objetivo mais rápido (figura 4.5 (c)). Já na figura 4.5 (d) o caminho mais curto apresenta maior concentração de feromônio. O número de linhas desenhados nos caminhos é proporcional à quantidade de feromônio depositado pelas formigas.

Estudos têm sido realizados no intuito de obter um melhor entendimento de como tais indivíduos tem sucesso exibindo um comportamento emergente complexo. O primeiro algoritmo inspirado em colônia de formigas tem origem no trabalho de Dorigo (1992), que propôs um sistema chamado de *ant system* para solucionar o problema do caixeiro viajante.

4.4 Algoritmos Baseados em Colônia de Formigas

O primeiro algoritmo inspirado em colônia de formigas tem origem no trabalho de (Dorigo *et al.* 1991b), que desenvolveram um sistema chamado de *ant system* para solucionar o problema do caixeiro viajante, e desde então vários algoritmos têm sido desenvolvidos, como: *Ant-Q* (Gambardella e Dorigo, 1995), *ant colony system* (Dorigo e Gambardella, 1997), *max-min ant system* (Stutzle e Hoos, 1997), *fast ant system* (Taillard, 1998), *antabu* (Roux *et al.* 1998; Roux *et al.* 1999; Kaji, 2001) e uma série de variantes desses algoritmos que podem ser encontrados em (Dorigo e Gambardella, 1997; Gambardella e Dorigo, 1997b; Michel e

Middendorf, 1998; Stutzle, 1998; Di Caro e Dorigo, 1998; Bullnheimer *et al.* 1999b; Gambardella *et al.* 1999a). Na sequência são sumarizados alguns dos principais algoritmos de colônia de formigas.

4.4.1 *Ant System*

O primeiro algoritmo de formiga foi desenvolvido por Dorigo *et al.* (1991b), referenciado como sistema de formiga. Embora seu desempenho não pareça satisfatório quando comparado com os demais algoritmos, sua importância foi propiciar o desenvolvimento de outros algoritmos baseados no paradigma de otimização por colônia de formigas.

No *ant system* cada formiga escolhe uma ação baseada no valor da probabilidade a cada iteração. Na intenção de evitar a seleção de ações indesejadas, cada formiga possui uma memória, que armazena os estados visitados, assegurando que o caminho seja visitado pela formiga somente uma vez. A probabilidade de escolha do caminho é proporcional ao feromônio e atratividade, que altera de acordo com a modelagem do problema. Após visitar todos os estados, a formiga deposita uma quantidade de feromônio nas arestas (*i*,*j*). Para calcular a probabilidade de escolher uma ação, cada formiga utiliza a regra da equação 4.12 (Dorigo *et al.* 1996):

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & \text{if } j \in N_i^k(t) \\ 0 & \text{if } j \notin N_i^k(t) \end{cases} \tag{4.12}$$

onde α e β são os parâmetros para indicar a importância do feromônio e da heurística respectivamente, τ_{ij} é a quantidade de feromônio na aresta *ij*, η_{ij} representa o efeito *a priori* do movimento de *i* para *j* (*e.g.*, atração ou a qualidade do movimento) influenciado por β. A concentração do feromônio τ_{ij} indica a importância das ações no passado, agindo como memória para os melhores movimentos. O conjunto N_i^k define os possíveis estados para cada formiga *k* no nó *i*.

Ao final de cada iteração, a taxa de evaporação diminui o valor do feromônio nas arestas. Isso evita que as formigas fiquem estacionadas em ótimos locais (ou seja, escolhendo arestas que representam o melhor valor local, onde no entanto, não melhoram a solução global),

diminuindo a probabilidade de escolher arestas que não foram utilizadas. A expressão que determina a variação do feromônio é apresentada na equação 4.13:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t) \tag{4.13}$$

onde ρ representa a taxa de evaporação com $\rho \in [0,1]$, simulando o esquecimento de decisões passadas das formigas. Quanto mais próximo do valor 1 está ρ, mais acelerada é a evaporação do feromônio. Portanto, quanto mais acelerado a evaporação, mais aleatória é a busca, resultando em maior exploração do espaço de busca. Para $\rho = 1$, a busca é completamente aleatória.

Assim que uma dada formiga completa um caminho, o feromônio de cada aresta é atualizado pela equação 4.14:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4.14}$$

com,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) \tag{4.15}$$

onde $\Delta\tau_{ij}^{k}(t)$ da equação 4.15 representa a quantidade de feromônio depositado pela formiga *k* no estado (*i*, *j*) no tempo *t*.

4.4.2 *Ant Colony System*

O *ant colony system* (ACS) que foi desenvolvido por Gambardella e Dorigo para melhorar o desempenho do algoritmo *ant system* (Dorigo e Gambardella, 1997). O ACS difere do *ant system* em aspectos como: (i) diferente regra de transição; (ii) diferente regra de atualização do feromônio; (iii) introdução de atualizações de feromônio locais; e (iv) uso de listas de estados candidatos para favorecer estados específicos. Cada uma dessas modificações é examinada na sequência.

A regra de transição do ACS, conhecida como critério de seleção *pseudo-aleatório-proporcional* foi desenvolvida para balancear as habilidades de exploração e explotação do algoritmo. Assim, a formiga *k* no estado *i* se move para o estado *j* empregando a regra de transição da equação 4.16:

$$j = \begin{cases} \arg\max_{u \in N_i^k(t)} \{\tau_{iu}(t)\eta_{iu}^{\beta}(t)\} & if\ q \le r_0 \\ J & if\ q > r_0 \end{cases} \tag{4.16}$$

onde $r \sim U(0,1)$, e $q_0 \in [0,1]$ é um parâmetro especificado; $J \in N_i^k(t)$ representa um estado selecionado aleatoriamente de acordo com a probabilidade calculada com a equação 4.17:

$$p_{iJ}^k(t) = \frac{\tau_{iJ}(t)\eta_{iJ}^{\beta}(t)}{\sum_{u \in N_i^k} \tau_{iu}(t)\eta_{iu}^{\beta}(t)} \tag{4.17}$$

no qual $N_i^k(t)$ representa o conjunto válido de estados a serem visitados.

A regra de transição da equação 4.17 induz as formigas a seguirem o caminho mais curto e estados com maior valor de feromônio. O parâmetro q_0 é usado para balancear a exploração e explotação; se $q \le q_0$ a formiga irá para o estado com maior feromônio; se $q > q_0$ a formiga poderá explorar novos estados. Desta forma, quanto menor o valor de q_0 maior a busca por novos estados.

Diferentemente do algoritmo AS, somente o melhor resultado (*i.e.*, a formiga que percorreu o menor caminho) é utilizado para atualização do feromônio nas arestas que pertencem ao melhor caminho. O feromônio é atualizado empregando a regra de atualização global da equação 4.18.

$$\tau_{ij}(t+1) = (1-\rho_1)\tau_{ij}(t) + \rho_1 \Delta\tau_{ij}(t) \tag{4.18}$$

onde,

$$\Delta\tau_{ij}(t) = \begin{cases} \dfrac{1}{f(x^+(t))} & if\ (i,j) \in x^+(t) \\ 0 & caso\ contrário \end{cases} \tag{4.19}$$

com $f(x)^+(t) = | x^+(t) |$, no caso de encontrar caminhos de menor custo.

A regra de atualização global do algoritmo ACS permite que formigas encontrem as melhores soluções. Tal estratégia favorece a explotação, sendo aplicada logo após as formigas construírem a solução.

Dorigo e Gambardella (1997) implementaram dois métodos para selecionar o caminho $x^+(t)$:

- *Iteration-Best*: onde $x^+(t)$ representa o melhor caminho encontrado durante a iteração corrente *t*, denotada como $\tilde{x}(t)$; e
- *Global-Best*: onde $x^+(t)$ representa o melhor caminho encontrado deste a primeira iteração do algoritmo, denotada como $\hat{x}(t)$.

Para a estratégia *Global-Best* o processo de busca é baseado em informações globais. O algoritmo de ACS utiliza a equação 4.20 como regra de atualização local:

$$\tau_{ij}(t) = (1-\rho_2)\tau_{ij}(t) + \rho_2\tau_0 \tag{4.20}$$

onde ρ_2 tem valor entre (0,1), e τ_0 é a constante com valor positivo.

No algoritmo de ACS também é modificado a maneira que os próximos estados são escolhidos. O conjunto de estados $N_i^k(t)$ é organizado para conter uma lista de estados candidatos. Esses estados são os preferidos e serão visitados inicialmente.

4.4.3 *Ant-Q*

Gambardella e Dorigo (1995) desenvolveram uma variante do algoritmo ACS onde a regra de atualização local foi inspirada no algoritmo *Q-learning* (apresentado na seção 4.2). No algoritmo *Ant-Q* o feromônio é denotado por *Ant-Q Value* (ou *AQ-Value*). O objetivo do *Ant-Q* é aprender *AQ-values* de tal forma que encontre boas soluções que favoreçam a tomada de decisão. A regra de transição utilizada para selecionar a próxima ação é mostrada na equação 4.21, onde μ_{su} denota o *AQ-values* do estado *s* e *u* no tempo *t*:

$$j = \begin{cases} \arg\max_{i \in N_s^k(t)} \{\mu_{si}^{\delta}(t)\eta_{si}^{\beta}(t)\} & \text{if } q \le q_0 \\ J & \text{caso contrário} \end{cases} \tag{4.21}$$

onde δ e β representam o valor de importância do *AQ-value* e da heurística respectivamente; q é um valor selecionado aleatoriamente com probabilidade uniforme em [0,1], quanto maior o valor de q_0 menor a probabilidade de escolher um estado aleatoriamente; J é uma variável

aleatória selecionada de acordo com a probabilidade dada pela função de *AQ-values* μ_{su} ($AQ(s,u)$); e η_{su} a informação heurística.

Três diferentes regras foram propostas para selecionar o valor da variável aleatória *J*:

- Regra de escolha de ações *pseudo-aleatória*: *J* é um estado selecionado aleatoriamente do conjunto de estados $N_s^k(t)$ de acordo com a distribuição uniforme;
- Regra de escolha de ações *pseudo-aleatória-proporcional*: *J* é selecionado de acordo com a distribuição apresentada na equação 4.22:

$$p_{sj}^k(t) = \begin{cases} \dfrac{\mu_{su}^{\delta}(t)\eta_{su}^{\beta}(t)}{\sum_{u \in N_s^k(t)} \mu_{si}^{\delta}(t)\eta_{si}^{\beta}(t)} & \text{if } j \in N_s^k(t) \\ 0 & \text{caso contrário} \end{cases} \tag{4.22}$$

e,

- Regra de escolha de ações *aleatória-proporcional*: com q_0= 0 na equação 4.21, o próximo estado será sempre selecionado aleatoriamente baseado na distribuição dada na equação 4.22.

Gambardella e Dorigo mostram que a melhor regra para a escolha das ações para o *Ant-Q* é o *pseudo-aleatória-proporcional* (considerando o problema do caixeiro viajante). Dessa forma, o *AQ-value* é aprendido utilizando a regra de atualização da equação 4.23, similar a do *Q-learning*, onde *y* representa o fator de desconto e α é a taxa de aprendizagem.

$$AQ(s,u) \leftarrow (1-\alpha).AQ(s,u) + \alpha.\left(\Delta AQ(s,u) + \gamma.\max_{i \in N_j^k(t)} AQ(j,i)\right) \tag{4.23}$$

No *Ant-Q* a equação de atualização 4.23 é aplicada para cada formiga *k* após o estado *j* ter sido selecionado, com $\Delta AQ(s,u)$= 0. O efeito é que o *AQ-value* associado à aresta (s,u) é reduzido pelo fator *y* cada vez que a aresta estiver na solução candidata (Gambardella e Dorigo, 1995).

Devido a similaridade do algoritmo *Ant-Q* com algoritmos de aprendizagem por reforço, ele será usado com a proposta apresentada no capítulo 5.

4.4.4 *Fast Ant System*

Taillard e Gambardella (1997) e Taillard (1998) desenvolveram o *fast ant system* (FANT), para resolver o problema de atribuição quadrática. A principal diferença entre o FANT e demais algoritmos de otimização por colônia de formigas é que o FANT usa somente uma formiga e a regra de atualização não utiliza estratégias de evaporação.

O uso de somente uma formiga reduz significativamente a complexidade computacional. O FANT usa como regra de transição a equação 4.21, com β= 0, onde nenhuma informação heurística é empregada. A regra de atualização do feromônio é definida pela equação 4.24:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \omega_1 \Delta \tilde{\tau}_{ij}(t) \; + \omega_2 \Delta \hat{\tau}_{ij}^{+}(t) \tag{4.24}$$

no qual w_1 e w_2 são os parâmetros para determinar o reforço relativo fornecido pela solução atual na iteração t e a melhor solução encontrada anteriormente. Os feromônios adicionados são calculados pelas equações 4.25 e 4.26:

$$\Delta \tilde{\tau}_{ij}(t) = \begin{cases} 1 & if\ (i,j) \in \tilde{x}(t) \\ 0 & caso\ contrário \end{cases} \tag{4.25}$$

e,

$$\Delta \hat{\tau}_{ij}(t) = \begin{cases} 1 & if\ (i,j) \in \hat{x}(t) \\ 0 & caso\ contrário \end{cases} \tag{4.26}$$

onde $\tilde{x}\ (\tau)$ e $\hat{x}\ (\tau)$ são respectivamente os melhores caminhos encontrados na iteração t e o melhor caminho global encontrado na busca.

Os feromônios são inicializados em $\tau_{ij}(0)$= 1. Assim que um novo $\hat{x}(t)$ é obtido, todos os feromônios são reinicializados em $t_{ij}(0)$= 1. Dessa forma, são exploradas as áreas próximas do melhor caminho global, $\hat{x}(t)$. Se no passo t, alguma solução é encontrada como a melhor solução global, o valor de w_1 é aumentado. Isso facilita a exploração diminuindo a contribuição $\Delta \hat{T}_{ij}(t)$, associada com o melhor caminho global.

4.4.5 *Antabu*

Roux *et al.* (1998), Roux *et al.* (1999) e Kaji (2001) adaptaram o *ant system* incluindo uma busca local para melhorar as soluções. Como procedimento de busca local, a regra de atualização global é alterada de forma que os feromônios gerados pelas formigas são depositados para construir um caminho proporcional a qualidade da solução. Cada formiga *k* usa a equação 4.27 para atualização do feromônio:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \left(\frac{\rho}{\int(x^{k}(t))}\right) \times \left(\frac{\int x^{-}(t)) - \int x^{k}(t))}{\int(\hat{x}(t))}\right) \tag{4.27}$$

onde $f(x^{-}(t)$ é o custo do pior caminho encontrado, $f(\hat{x}(t))$ é o custo do melhor caminho encontrado, e $f(x^{k}(t)$ é o custo do caminho encontrado pela formiga *k*. A equação 4.27 é aplicada por cada formiga k para cada ligação $(i, j) \in x^{k}(t)$.

4.4.6 *AS-rank*

Bullnheimer *et al.* (1999a) propuseram algumas modificações para o *ant system*, tais como:

i) permitir que somente a melhor formiga atualize os feromônios;

ii) usar formigas elitistas; e

iii) permitir que as melhores formigas sejam selecionadas e então elencadas para atualizarem os feromônios.

No *AS-rank*, a regra de atualização global é alterada conforme a equação 4.28:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + n_e \Delta\hat{\tau}_{ij}(t) + \Delta\tau_{ij}^{r}(t) \tag{4.28}$$

onde,

$$\Delta\hat{\tau}_{ij}(t) = \frac{Q}{\int(\hat{x}(t))} \tag{4.29}$$

no qual $\hat{x}(t)$ é o melhor caminho. Se n_e é usado pelas formigas elitistas e n_k são as formigas ordenadas $f(x^{1}(t)) \leq f(x^{2}(t)) \leq \ldots \leq f(x^{nk}(t)$, logo:

$$\Delta \tau_{ij}^{\sigma}(t) = \sum_{\sigma=1}^{ne} \Delta \tau_{ij}^{\sigma}(t) \tag{4.30}$$

onde,

$$\Delta \tau_{ij}^{\sigma}(t) = \begin{cases} \dfrac{(\eta_e - \sigma)Q}{\int (x^{\sigma}(t))} & \text{if } (i,j) \in x^{\sigma}(t) \\ 0 & \text{caso contrário} \end{cases} \tag{4.31}$$

no qual σ na equação 4.31 indica a classificação da formiga. Esta estratégia elitista difere da AS, na qual a atualização das formigas elencadas é diretamente proporcional a sua classificação: quanto melhor sua classificação (*i.e.*, menor σ) maior sua contribuição.

4.4.7 Resoluções com Algoritmos de Colônia de Formigas

Algoritmos de colônia de formigas têm sido aplicados em diversas classes de problemas de otimização, como: roteamento de veículos (Gambardella *et al.* 1999a); atribuição quadrática (Gambardella *et al.* 1999b); atribuição bi-quadrática de recursos (Taillard, 1998), coloração de grafos (Costa e Hertz, 1997), circuitos digitais (Abd-El-Barr *et al.* 2003), circuitos lógicos (Coello *et al.* 2002), dentre outros.

Em problemas de roteamento de veículos agentes devem visitar um conjunto predefinido de localizações, no qual uma função objetivo depende da ordenação dos locais visitados. O roteamento de veículos requer a determinação de um conjunto ótimo de rotas para que um conjunto de veículos atenda a demanda. Mazzeo e Loiseau (2004) propuseram um roteamento de veículos capacitados, onde existe limite de peso e capacidade de volume que cada veículo pode transportar. Os autores utilizaram um algoritmo de colônia de formigas para o roteamento de veículos baseado na técnica de metaheurística, introduzida em (Dorigo, 1992). O objetivo é atender o conjunto de pontos de demanda, localizados nos vértices do grafo G=(N,A), de modo a minimizar o comprimento total das rotas dos veículos.

Bell e McMullen (2004) aplicaram um método de otimização de colônia de formigas para estabelecer um conjunto de problemas de roteamento de veículos. Foram simulados processos de tomada de decisão, onde o algoritmo de colônia de formigas foi alterado para permitir a busca de múltiplas rotas. O uso de múltiplas colônias de formigas fornece soluções competitivas, especialmente em problemas complexos. Quando comparado com outros métodos, o tempo computacional é favorável.

Em problemas de atribuição, a tarefa é atribuir um conjunto de itens (objetos, atividades) para um dado número de recursos (locações, agentes) determinados. Atribuições podem ser representadas como um mapeamento de um conjunto I para um conjunto J, e uma função objetiva para minimizar as atribuições. Gámez e Puerta (2002) propuseram uma nova maneira de lidar com problemas de triangulação de gráficos. O problema é centralizado em aplicações de otimização por colônia de formigas, na qual heurísticas são utilizadas para as atribuições. O uso de heurísticas melhora os resultados obtidos por outras técnicas, tanto em precisão como na eficiência. Algoritmos genéticos foram utilizados por (Gámez e Puerta, 2002) para testar os algoritmos de otimização, acelerando o processo de busca devido às boas soluções na fase inicial da população.

Lim *et al.* (2006) desenvolveram uma heurística para resolver problemas de largura de banda que utilizou o método de busca subida da montanha (*hill climbing*) guiada por algoritmos de colônia de formigas. O método foi comparado com outras abordagens de otimização por colônia de formigas, mostrando que busca local eficiente, combinada com mecanismo de busca global pode produzir resultados competitivos. Dois métodos construtivos foram utilizados com o algoritmo de colônia de formigas. Maniezzo e Carbonar (2000) consideram o problema em atribuir frequências de rádio entre a estação base e transmissores móveis, na intenção de minimizar a interferência global sobre uma determinada região. Como o problema é *NP*-completo, foi aplicada uma heurística baseada em otimização por colônia de formigas. Resultados experimentais mostram a eficiência da abordagem proposta.

O problema de atribuição em células é essencial para o desenvolvimento de serviços de comunicação pessoal. Shyu *et al*. (2006) desenvolveram um algoritmo de otimização por colônia de formigas para resolver o problema em serviços de comunicação pessoal. O problema é modelado em forma de combinação em um grafo ponderado bipartido dirigido, de modo que formigas artificiais possam construir seus caminhos. Experimentos foram realizados para captar o comportamento das formigas em problemas de otimização e analisar o desempenho do algoritmo de colônia de formigas.

Annaluru *et al*. (2004) propuseram um algoritmo de colônia de formigas para encontrar localizações ótimas e classificar capacitores em rede de distribuição em compensação de potência reativa. A abordagem é multinível no qual duas tabelas de feromônios são mantidas pelo algoritmo. Formigas geram soluções estocásticas, baseadas nas tabelas de feromônio que são atualizadas periodicamente, de maneira que os feromônios acumulados melhorem a solução atual ao longo do tempo. Resultados obtidos pelo algoritmo proposto foram comparados

com outras técnicas, mostrando que a abordagem pode ser aplicada em problemas de classificação.

Em problemas de alocação de recursos, o objetivo é alocar recursos para atividades de forma que o custo se torne ótimo. Lee e Lee (2005) desenvolveram um algoritmo de busca híbrido com heurística para problemas de alocação de recursos encontrados na prática. O algoritmo proposto têm as vantagens dos algoritmos genético e colônia de formigas, que permitem explorar o espaço de busca pela melhor solução. Resultados parecem mostrar que devido às propriedades dos algoritmos genético e colônia de formigas, a abordagem híbrida supera outros algoritmos existentes.

Em problemas de roteamento de redes o objetivo é encontrar caminhos com menor custo na rede. Se os custos da rede são fixos, então o problema de roteamento de redes é reduzido a um conjunto de caminhos de custo mínimo, que pode ser resolvido usando algoritmos de tempo polinomial. Bean e Costa (2005) apresentam uma técnica de modelagem analítica para o estudo de uma nova classe de algoritmos de roteamento de rede adaptativo, a qual é inspirada na resolução de problemas emergentes observadas em colônias de formigas. Esta classe de algoritmos utiliza agentes chamados de *antlike* que percorrem a rede e constroem coletivamente políticas de roteamento. Resultados indicam que o algoritmo tem bom desempenho em relação às mudanças em tempo real em demandas de tráfego e condições da rede.

Su *et al*. (2005) propuseram um algoritmo de busca de colônia de formigas para resolver problemas de reconfiguração de redes para reduzir a perda de energia elétrica. O problema de reconfiguração de rede de um sistema de distribuição da companhia de energia de Taiwan foi resolvido usando o método proposto. Tal método foi comparado com um algoritmo genético metaheurístico (computação evolutiva) e a metaheurística têmpera simulada (*simulated annealing*). Resultados numéricos mostram que o método proposto por eles é melhor do que os demais métodos, sob as condições de modelagem apresentadas.

Além dos trabalhos discutidos, há ainda outros métodos que apresentam características interessantes não presentes em outras implementações de otimização por colônia de formigas. Tsai *et al*. (2004) propuseram um novo algoritmo de colônia de formigas com *different favor* para resolver problemas de agrupamento de dados. O algoritmo possui as seguintes estratégias: i) adota conceitos de têmpera simulada para as formigas diminuírem o número de estados visitados, e ii) utiliza estratégia de torneio de seleção para escolher o caminho. O método é comparado com a técnica de mapas auto-organizáveis com *K-means* e algoritmos genéticos. O algoritmo parece eficiente e preciso em conjuntos de dados com alta dimensão.

Guntsch e Middendorf (2001) propuseram uma técnica para melhorar a solução quando alterações ocorrem no ambiente, aplicando procedimentos de busca local para as soluções. Alternativamente, estados afetados pela mudança são retirados da solução, conectando o estado predecessor e sucessor do estado excluído. Dessa forma, novos estados (não usados ainda na solução) são inseridos na solução. O novo estado é inserido na posição que causa o custo mínimo ou diminui o custo mais alto (dependendo do objetivo) no ambiente.

Sim e Sun (2002) usaram múltiplas colônias de formigas, onde uma colônia é repelida pelo feromônio de outras colônias favorecendo a exploração quando o ambiente é alterado. Outras técnicas para tratar a dinâmica no ambiente alteram a regra de atualização do feromônio para favorecer a exploração. Por exemplo, Li e Gong (2003) modificaram as regras de atualização local e global do *ant colony system*. A regra de atualização local foi alterada conforme equação 4.32:

$$\tau_{ij}(t+1) = (1-\rho_1(\tau_{ij}(t)))\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4.32}$$

onde $\rho_1(\tau_{ij})$ é uma função de τ_{ij}, *e.g.*:

$$p_1(\tau_{iJ}) = \frac{1}{1+\mathrm{e}^{-(\tau_{ij}+\theta)}} \tag{4.33}$$

onde $\theta > 0$.

A dinâmica da evaporação faz com que valores elevados de feromônios fossem diminuídos. Assim, quando o ambiente se altera e a solução não é a melhor, a concentração de feromônio nas arestas correspondentes diminui ao longo do tempo. A atualização global é feita similarmente, mas consideram somente a melhor e a pior solução global (equação 4.34):

$$\tau_{ij}(t+1) = (1-\rho_2(\tau_{ij}(t)))\tau_{ij}(t) + \gamma_{ij}\Delta\tau_{ij}(t) \tag{4.34}$$

onde,

$$\gamma_{i,j} = \begin{cases} +1 & se\,(i,\,j)\,é\,a\,melhor\,solução\,global \\ -1 & se\,(i,\,j)\,é\,a\,pior\,solução\,global \\ 0 & caso\,contrário \end{cases} \tag{4.35}$$

Uma regra similar de atualização global também foi usada em (Lee *et al.* 2001a; Lee *et al.* 2001b). Guntsch e Middendorf (2001) propuseram três regras de atualização do feromônio para ambientes dinâmicos. O objetivo das regras é encontrar um equilíbrio ótimo da recompo-

sição de informações, permitindo explorar novas soluções enquanto mantém informações suficientes de processos de buscas passadas, para acelerar o processo para encontrar uma solução. Para cada estratégia, $y_1 \in [0,1]$ é calculado e o feromônio reinicializado, conforme equação 4.36:

$$\tau_{ij}(t+1) = (1-\gamma_{ij})\tau_{ij} + \gamma_i \frac{1}{nG-1} \tag{4.36}$$

onde *nG* é o número de estados na representação. As seguintes estratégias foram propostas:

- Estratégia de recomeço: para essa estratégia é usada a equação 4.37:

$$\gamma_1 = \lambda_R \tag{4.37}$$

onde $\lambda_R \in [0,1]$ é referido como parâmetro de estratégica específica, onde as alterações no ambiente não são consideradas.

- η-estratégia: informação heurística usada para decidir o grau de valor dos feromônios atualizados (equação 4.38):

$$\gamma_i = \max\{0, d_{ij}^{\eta}\} \tag{4.38}$$

na qual,

$$d_{ij}^{\eta} = 1 - \frac{\overline{\eta}}{\lambda_\eta \eta_{ij}}, \lambda_\eta \in [0,\infty) \tag{4.39}$$

e,

$$\overline{\eta} = \frac{1}{nG(nG-1)} \sum_{i=1}^{nG} \sum_{j=1, j\neq i} \eta_{ij} \tag{4.40}$$

Nesse caso, y_i é proporcional à distância do estado alterado, e a atualização é realiza em todas as arestas incidentes.

- τ-estratégia: o valor do feromônio é usado para atualizar as arestas próximas ao estado alterado (equação 4.41):

$$\gamma_i = \min\{1, \lambda_T d_{ij}^{\eta}\},\ \lambda_T \in [0, \infty) \tag{4.41}$$

na qual,

$$d_{ij}^{\tau} = \max_{N_{ij}} \left\{ \prod_{(x,y) \in N_{ij}} \frac{\tau_{xy}}{\tau_{\max}} \right. \tag{4.42}$$

e N_{ij} é o conjunto do percurso de *i* até *j*.

4.5 Considerações Finais

Neste capítulo foram estudadas as principais características da aprendizagem por reforço e baseada em enxames. Podem-se observar diversas semelhanças desses conceitos, sendo que um agente aprende interagindo em um ambiente baseado no comportamento individual ou social. Quando a interação entre os agentes melhora o comportamento coletivo, então a tendência para se reproduzir uma relação entre eles é reforçada. Observou-se que a aprendizagem por reforço é uma metodologia utilizada por diversos algoritmos. Uma maneira de formalizar esses algoritmos é utilizando conceitos de processos decisórios de *Markov*, onde o formalismo inclui um conjunto de estados do ambiente, um conjunto de ações do agente, um conjunto de transições de estados e uma função de recompensas.

Foram citados alguns dos principais algoritmos baseados em recompensas. Algoritmos como o *Q-learning* e o *Ant-Q* possuem propriedades em comum, pois buscam aprender uma política de maneira iterativa quando o modelo do sistema não é conhecido, a partir de recompensas que podem ser socializadas com os demais agentes. Neste trabalho defende-se a ideia de que essas recompensas podem ser socializadas com modelos específicos de compartilhamento de recompensas e princípios de redes sociais, onde uma estrutura social dinâmica pode ser identificada e utilizada para melhorar a coordenação dos agentes. Esse é o princípio fundamental da metodologia apresentada no capítulo a seguir.

Capítulo 5

Enfoque Proposto

Observou-se nos capítulos anteriores que muitos métodos de coordenação utilizam algoritmos de aprendizagem por reforço para coordenar as tarefas dos agentes. Muitas atividades precisam ser realizadas em conjunto, pois um único agente não concentra todos os recursos e habilidades necessárias para satisfazer o objetivo global. As interações entre os agentes e as recompensas geradas estabelecem o comportamento individual e social. As interações acabam por formar uma estrutura social, na qual a sociabilidade interfere no comportamento dos agentes que interagem a fim de executar tarefas em comum. Um dos objetivos desse trabalho é desenvolver modelos sociais para compartilhamento de recompensas sociais. Agentes que interagem compartilhando recompensas devem dispor de modelos específicos que permitam melhorar o comportamento global do sistema. Isso vai ao encontro do segundo objetivo, que propõe utilizar a estrutura social construída com as relações dos indivíduos de um sistema multiagente.

A metodologia apresentada neste capítulo está dividida em três partes:

a) estudo do impacto de recompensas sociais em algoritmos de aprendizagem por reforço como forma de aprimorá-los (seção 5.1);
b) estudo das características e do comportamento do *Ant-Q* em diferentes cenários (seção 5.2) de forma a esclarecer os aspectos que podem ser melhorados com o auxílio de recompensas e teorias sociais. Também propomos nessa seção estratégias que podem ser usadas em estruturas sociais dinâmicas, aproveitando as recompensas de políticas aprendidas;
c) estudo do impacto das teorias sociais como meio para se estabelecer fundamentos teóricos e computacionais para um método social de otimização baseado no *Ant-Q* (seção 5.3), uma vez que esta técnica de coordenação multiagente também segue

princípios de aprendizagem por reforço e os experimentos apresentados na seção 5.2 demonstraram o potencial e a viabilidade dessa proposta.

5.1 Impacto das Recompensas em Aprendizagem por Reforço

Foi discutido no capítulo 4 que algoritmos de aprendizagem por reforço, como o *Q-learning*, podem ser utilizados para a descoberta de políticas de ação com um único agente, explorando repetidamente o espaço de estados. A política de ação determina a sequência de ações que devem ser executadas no ambiente, gerando um processo decisório de *Markov*. Usar um único agente tende a ser ineficiente em problemas onde o espaço de estados é grande. Nestes casos, aprendizagem por reforço com múltiplos agentes tem se mostrado uma alternativa interessante (Ribeiro, 2008a). Na aprendizagem por reforço, cada agente pode ter uma política parcial (*i.e.*, cada agente pode ter acesso à parte do conhecimento global (Weiss, 1996)), e o objetivo é interagir para formar relações com os demais agentes, no intuito de melhorar o conhecimento sobre o ambiente e melhorar a qualidade global da solução.

Diferentes abordagens foram propostas para aprendizagem por reforço com múltiplos agentes (Mataric, 1998; Chapelle *et al.* 2002; Hadad e Kraus, 2002; Soh e Luo, 2003; Schermerhorn e Scheutz, 2006; Ribeiro, 2008a), no entanto, elas geralmente apresentam problemas de convergência por não possuírem um modelo explícito micro e macro para a coordenação dos agentes. Para melhorar a coordenação em abordagens de aprendizagem por reforço com múltiplos agentes são propostos modelos de interação que permitem o compartilhamento de recompensas geradas. Quando os agentes interagem, políticas de ação são geradas para indicar uma possível solução para o problema. Tais políticas são formadas por valores que determinam as ações dos agentes. Quando esses valores são compartilhados pelos agentes, ocorre a aprendizagem por **recompensas compartilhadas**, que determinam a intensidade da relação entre os agentes no sistema.

Os modelos de compartilhamento de recompensas propostos neste trabalho definem as melhores recompensas do sistema a partir das seguintes estratégias:

i) recompensas compartilhadas em episódios pré-determinados;

ii) recompensas compartilhadas a cada ação, a partir de uma regra de transição baseada na própria política de ação; e

iii) recompensas compartilhadas de forma local e global, conforme a configuração dos parâmetros do algoritmo *Q-learning*.

Antes de discutir profundamente os modelos e os resultados experimentais, é formalizado um *framework* implementado para a avaliação dos algoritmos de aprendizagem.

5.1.2 Aprendizagem por Recompensas Partilhadas

A aprendizagem por reforço com múltiplos agentes baseada em recompensas compartilhadas pode produzir um conjunto refinado de comportamentos obtidos a partir das ações tomadas. Parte do conjunto de comportamentos (*i.e.*, uma política global) é compartilhada pelos agentes por meio de uma política de ação parcial (Q_i). Geralmente, tais políticas parciais contêm informações (valores de aprendizagem) incompletas sobre o ambiente, mas com um modelo para compartilhar as recompensas, essas podem ser integradas para maximizar a soma das recompensas parciais obtidas ao longo da aprendizagem. Quando políticas $Q_1,\ldots,Q_x$ são integradas, é possível formar uma nova política, denominada de política de ação baseada em recompensas partilhadas $\partial = \{Q_1,....,Q_x\}$, na qual $\partial(s,a)$ é uma tabela que denota as melhores recompensas adquiridas pelos agentes durante o processo de aprendizagem.

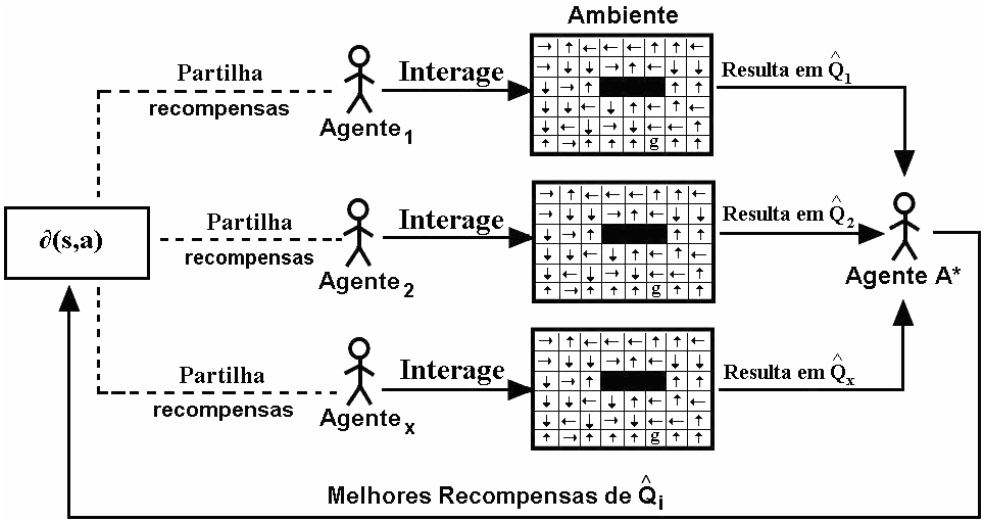


Figura 5.1: Interação com informações partilhadas

A figura 5.1 mostra como os agentes trocam informações ao longo das interações. Empregando o algoritmo *Q-learning*, um Agente$_i$ gera e armazena as recompensas em $\hat{Q}_i$. Quando o agente A* recebe as recompensas o seguinte procedimento é realizado: quando o Agente$_i$ alcança o estado objetivo g a partir do estado inicial $s \neq g$ com um caminho de menor custo, o agente usa um modelo para compartilhar as recompensas com os demais agentes. Os valores de aprendizagem de uma política parcial Q_i podem ser utilizados para atualizar a política glo-

bal $\partial(s,a)$ disponível, interferindo posteriormente na forma como os demais agentes atualizam seus conhecimentos e interagem com o ambiente.

O algoritmo da figura 5.2 apresenta a função que compartilha as recompensas dos agentes. Essa tarefa pode ser realizada de três formas e são discutidas na subseção 5.1.3, sendo que todas elas utilizam internamente um algoritmo de aprendizagem por reforço (*Q-learning*). As melhores recompensas de cada agente são enviadas para a $\partial(s,a)$, formando uma nova política com as melhores recompensas adquiridas pelos agentes $I = \{i_1,...i_x\}$, que na sequência podem ser socializadas com os demais agentes. Para estimar $\partial(s,a)$, é usada uma função custo, que encontra o caminho de menor custo do estado inicial ao estado objetivo em uma dada política. A descoberta desse caminho é realizada com o algoritmo A* que produz um modelo generativo para administrar a política que maximiza a recompensa total esperada, *i.e.*, política ótima, de acordo com a metodologia apresentada em (Ribeiro *et al.* 2006c).

Os seguintes elementos são necessários para a compreensão e formalização dos modelos sociais de aprendizagem por reforço propostos:

- Um conjunto de estados $S = \{s_1,...,s_m\}$;
- Um conjunto de agentes $I= \{i_1,...,i_x\}$;
- Políticas de ação parcial $\{Q_1,....,Q_x\}$, onde Q_i representa a política parcial do agente i;
- Uma função de recompensa st(S) $\rightarrow ST$, onde $ST= \{-1; -0,4; -0,3; -0,2; -0,1\}$;
- Um instante de tempo de valor discreto $t= 1,2,3,...,n$;
- Um episódio de tempo c onde $c < n$;
- Um conjunto de ações $A= \{a_1,...,a_k\}$, onde cada ação é executada no tempo t;
- Uma tabela de aprendizagem $\hat{Q} : (S \times A) \rightarrow \Re$, que define uma política Q;
- Um conjunto de modelos de compartilhamento de recompensas $M= \{$*discreto*, *contínuo*, *dirigido por objetivo*$\}$ (vide pp. 111/2);
- Uma função custo: $custo(s, g) = \sum_{s \in S}^{g} 0.1 + \sum_{s \in S}^{g} st(S)$ usada para calcular o custo de um episódio (caminho do estado inicial s até o estado objetivo g) baseado na política atual;
- Uma função que define o modelo de partilha f: $(t \times M \times I \times S \times A) \rightarrow \partial(s,a)$, onde t é a condição de parada do modelo;
- Uma política ótima objetivo Q^* estimada com um algoritmo supervisor (A^*);

```
Algoritmo aprendizagem_social (I, modelo)
Tabela de aprendizagem: Qi, Q*, ∂
01  ∂(s,a) ← 0 ;
02  Para cada agente i ∈ I faça:
03     Para cada estado s ∈ S faça:
04        // inicialização dos valores de aprendizagem
        Para cada ação a ∈ A faça:
05        Q̂_i(s,a) ← 0 ;
06      Fimpara
07    Fimpara
08  Fimpara
09  t ← 0;
10  Para cada agente i ∈ I faça:
11    Enquanto não f(t, modelo, Q*, Qi, ∂(s,a)) repita:
12       t ← t + 1;
13       Escolha estado s ∈ S, ação a ∈ A
           Atualize:
14              V ← γ max_{a_{t+1}} Q̂_t(s_{t+1}, a_{t+1}) − Q̂_t(s_t, a_t) ;
                Q̂_{t+1}(s_t, a_t) ← Q̂_t(s_t, a_t) + α[R(s_t, a_t) + V] ;
15    Fimenquanto
16  Fimpara
17  Para cada agente i ∈ I faça:
18    Q̂_i(s,a) ← ∂(s,a) ;
19  Fimpara
20  Fim
```

Figura 5.2: Algoritmo de aprendizagem por reforço social

O algoritmo da figura 5.2 deve ser interpretado da seguinte forma:

- Linha 2-8: Inicialização da $\hat{Q}_i(s,a)$;
- Linha 10: Interação dos agentes i ∈ I;
- Linha 11: A função *f* seleciona um modelo de partilha; no qual < *t, modelo,* Q_i*, s, a* > são os parâmetros, onde *t* é a iteração corrente, modelo ∈ {discreto, contínuo, dirigido por objetivo}, *s* e *a* são o estado e a ação escolhidos respectivamente da política Q_i;
- Linha 14: Para cada par estado-ação é empregada a regra de atualização que calcula os valores de recompensas;
- Linha 17-18: $\hat{Q}_i$ do agente $i \in I$ é atualizado com ∂(s,a);

```
Algoritmo f (t, modelo, Q*, Qi, ∂(s,a))
C: número de episódios
 01  Escolha modelo:
 02   Caso "discreto":
 03     Se t mod C = 0 então
 04       ∂(s,a) = atualiza_política(Q*, Qi, ∂(s,a))
 05     Fimse
 06   Caso "contínuo":
 07     r ← Σ_{i=1}^{x} Q̂_i(s,a) ;
 08     Q̂_i(s,a) ← r ;
 09     ∂(s,a) = atualiza_política(Q*, Qi, ∂(s,a))
 10   Caso "dirigido_por_objetivo":
 11     Se s = g então
 12       r ← Σ_{i=1}^{x} Q̂_i(s,a) ;
 13       Q̂_i(s,a) ← r ;
 14       ∂(s,a) = atualiza_política(Q*, Qi, ∂(s,a))
 15     Fimse
 16  Fimescolha
 17  Retorne(∂(s,a))
```

Figura 5.3: Modelos de compartilhamento de recompensas

```
Função atualiza_política(Q*, Qi, ∂(s,a))
 01    Para cada estado s ∈ S faça:
 02      Se custo(Qi,s) ≤ custo(Q*,s) então
 03          ∂(s,a) ← Q̂_i(s,a) ;
 04      Fimse
 05    Fimpara
 06  Fimpara
 07  Retorne(∂(s,a))
```

Figura 5.4: Atualiza política

Para auxiliar a compreensão dos pseudocódigos, é ilustrado na figura 5.5 o diagrama de atividades que utiliza os algoritmos 5.2, 5.3 e 5.4.

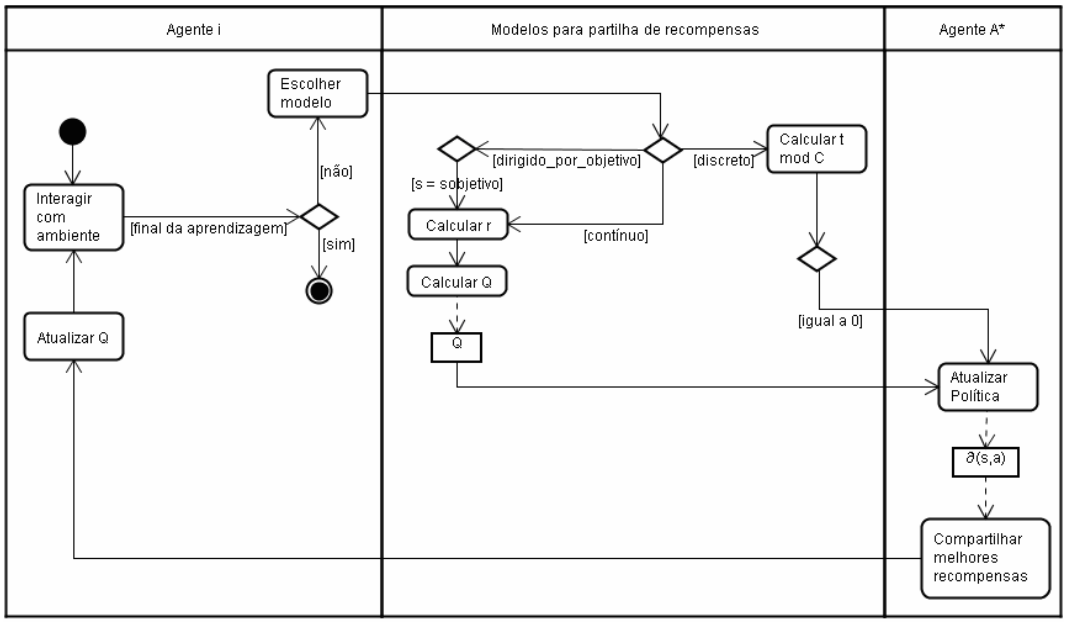


Figura 5.5: Diagrama de atividade do processo de aprendizagem

5.1.3 Modelos de Compartilhamento de Recompensas para Aprendizagem Multiagente

Os seguintes modelos de compartilhamento de recompensas são discutidos nesta subseção:

a) **discreto**: recompensas compartilhadas em determinados episódios;

b) **contínuo**: recompensas compartilhadas a cada ação; e

c) **dirigido por objetivo**: compartilha a soma das recompensas quando o agente alcança o estado objetivo.

Modelo Discreto: com o modelo discreto o agente acumula as recompensas obtidas a partir de suas ações ao longo das iterações (por exemplo, c iterações). No final da iteração c cada agente informa os valores da $\hat{Q}_i$ para a $\partial(s,a)$. Se o valor da recompensa é adequado, *i.e.* se melhora a eficiência dos outros agentes para o mesmo estado (linha 3 do algoritmo 5.4) os agentes socializam essas recompensas (linha 18 do algoritmo 5.2). Caso a recompensa seja útil somente para o agente que gerou a recompensa, o agente continuará usando tais recompensas com o objetivo de acumular novos valores que possam ser compartilhados nas próximas iterações. É observado que empregando o modelo discreto os agentes são capazes de compartilhar as melhores recompensas, dado que haja uma quantidade suficiente de iterações para acumular

boas recompensas. A soma das recompensas obtidas na iteração é geralmente capaz de melhorar a convergência de ∂.

Modelo Contínuo: com o modelo contínuo os agentes se relacionam compartilhando o valor do reforço obtido a cada transição $\tau^{a}_{s,s'}$. O reforço obtido é determinado pela ação baseado na política de ação. A aprendizagem ocorre da seguinte maneira: cada ação do agente gera um valor de reforço (linha 7 algoritmo 5.3). O objetivo é acumular altas recompensas em $\hat{Q}_i$ que possam ser compartilhadas no final do processo da aprendizagem. (linha 18 do algoritmo 5.2).

A intenção com esse modelo é mostrar que recompensas podem ser acumuladas mesmo quando geradas por transições de diferentes políticas, demonstrando que recompensas adquiridas de várias políticas geradas separadamente podem gerar uma política Q*.

Modelo Dirigido por Objetivo: diferentemente do modelo discreto, a cooperação ocorre quando o agente alcança o estado objetivo (linha 11 algoritmo 5.3). Neste caso, o agente interage no ambiente com o objetivo de acumular as maiores recompensas. Isso é necessário porque com esse modelo o agente compartilha suas recompensas em diferentes episódios. Portanto, essa estratégia usa como heurística a rápida acumulação de recompensas adquiridas pelos agentes durante a aprendizagem.

Quando o agente alcança o estado objetivo, o valor das recompensas adquiridas é enviado para a $\partial(s,a)$. Se o valor da recompensa do estado melhora a eficiência global, então os agentes compartilham tais recompensas. Isso mostra que mesmo compartilhando recompensas baixas e não satisfatórias do início da aprendizagem, o agente é capaz de aprender, sem prejudicar a convergência global.

5.1.3.1 Resultados Experimentais com os Modelos Discreto, Contínuo e Dirigido por Objetivo

Os modelos sociais de aprendizagem por reforço foram avaliados a partir de um ambiente artificial construído com este objetivo. O ambiente de simulação utilizado para avaliação dos modelos é constituído por um espaço de estados onde há um estado inicial ($s_{inicial}$), um estado objetivo (g) e um conjunto de ações A= {↑ (para frente), → (para a direita), ↓ (para trás), ← (para a esquerda)}. Um estado s é um par (X,Y) com coordenadas de posições nos eixos X e Y respectivamente. Em outras palavras, o conjunto de estados S representa um mapa de uma cidade. No ambiente há uma função status st : S → ST que mapeia os estados e situações de tráfego (recompensas) onde ST = {–0,1 (livre); –0,2 (pouco congestionado); –0,3

(congestionado ou desconhecido); –0,4 (muito congestionado); –1 (bloqueado); 1,0 (*g*)}. Após cada movimento do agente (transição) de um estado *s* para o estado *s'*, o agente sabe se sua ação foi positiva ou negativa por meio das recompensas atribuídas. A recompensa para a transição $\tau^{a}_{s,s'}$ é *st*(*s'*). A figura 5.6 mostra uma representação simplificada de um ambiente com uma política π.

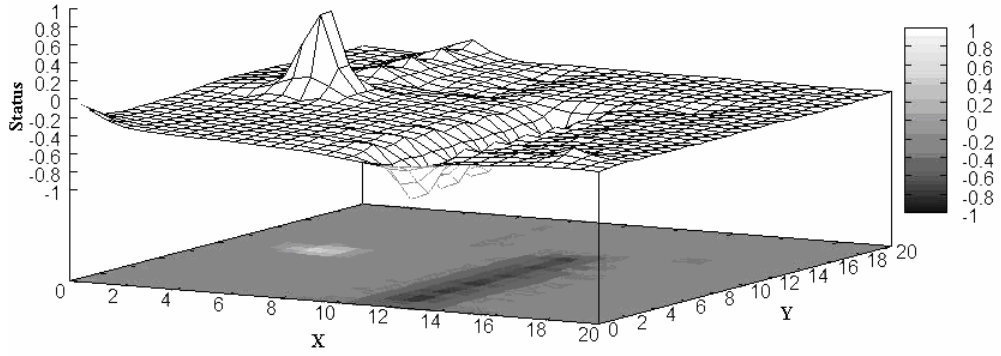


Figura 5.6: Exemplo de um ambiente com 400 estados. Os agentes são posicionados aleatoriamente no ambiente e possuem campo de profundidade visual de 1

Os experimentos foram realizados com agentes que empregam o algoritmo *Q-learning* original e os modelos sociais de recompensas descritos anteriormente. Os seguintes parâmetros foram usados: γ= 0,9 e α= 0,2. De 1 a 10 agentes foram usados com o objetivo de avaliar o impacto no ambiente da interação social produzida com os modelos.

Na intenção de observar a convergência em ambientes com muitos estados, os algoritmos foram executados em ambientes com 400 estados (20×20). É lembrado que um número de estados *S* pode gerar um grande espaço de soluções, na qual o número de políticas possíveis é $|A|^{|S|}$. Quinze ambientes com configurações diferentes foram gerados arbitrariamente (figura 5.7). O processo de aprendizagem em cada ambiente foi repetido quinze vezes (quantidade de amostras) para avaliar a variação na eficiência, que pode ocorrer devido aos valores gerados na aprendizagem com ambientes diferentes. Os valores apresentados correspondem à média de todos os experimentos gerados. Pôde-se observar que a eficiência dos modelos não foi significativamente afetada (±2,15%) quando um número maior de iterações foi utilizado.

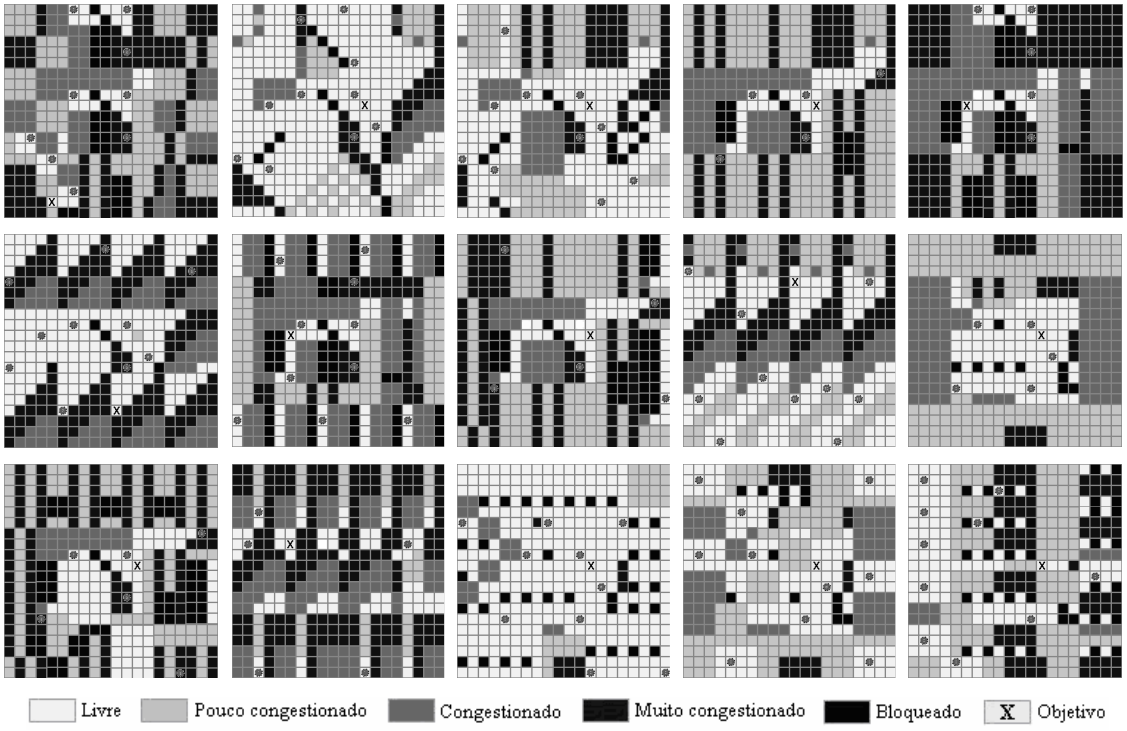


Figura 5.7. Ambientes usados nas simulações

Inicialmente, a $\partial(s,a)$ foi avaliada usando o modelo discreto com as políticas de ação parcial. Esse modelo converge para a política ótima Q* porque os reforços adquiridos pelos agentes são gerados em iterações pré-definidas e geralmente acumulam valores de reforços satisfatórios que levam a uma boa convergência. Pode-se observar na figura 5.8 que com o modelo discreto a $\partial(s,a)$ converge para Q*, pois há um intervalo suficiente de iterações para acumular as melhores recompensas (região R1).

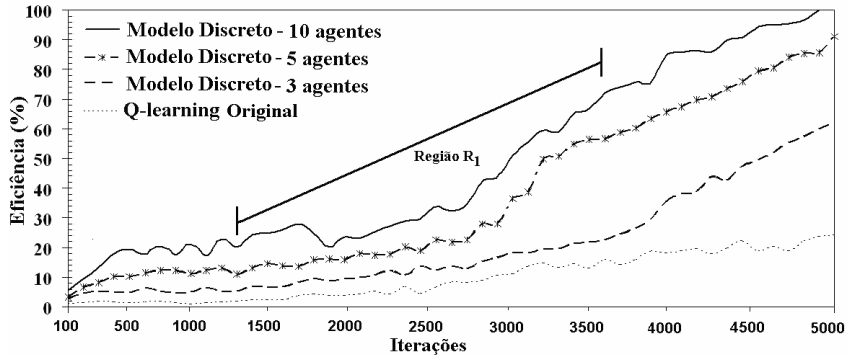


Figura 5.8: Modelo discreto

Já a figura 5.9 ilustra o desempenho do sistema usando o modelo contínuo. A $\partial(s,a)$ do modelo contínuo é capaz de acumular bons valores de reforços em poucas iterações. Pode-se

observar na figura 5.9 que depois de algumas iterações, o desempenho da ∂ diminui (região R2). Isso ocorre porque os estados mais próximos do estado objetivo começam a acumular reforços com valores altos, caracterizando um máximo local, penalizando o agente que ao longo do tempo não é capaz de visitar outros estados. Esse problema pode ser minimizado empregando a estratégia ε-greedy (Sutton e Barto, 1998). Quando a probabilidade ε é máxima, somente as melhores ações são escolhidas, limitando o espaço de possibilidades. Portanto, a exploração de política ε-greedy seleciona as ações aleatoriamente, com probabilidade ε e as melhores ações com probabilidade $1 - \varepsilon$. Essa abordagem foi discutida em (Ribeiro *et al.* 2006a; Ribeiro, 2006c).

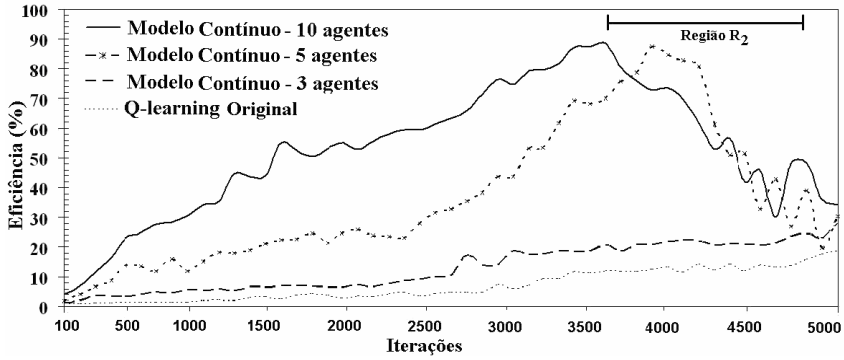


Figura 5.9: Modelo contínuo

A figura 5.10 ilustra o desempenho dos agentes com o modelo dirigido por objetivo. Com esse modelo o agente compartilha o aprendizado em um número variável de episódios e a cooperação ocorre quando o agente alcança o estado objetivo. A $\partial(s,a)$ gerada é capaz de acumular bons valores de reforços, dado que haja uma quantidade de iterações suficiente para acumular valores de recompensas satisfatórias. Nas iterações iniciais, o desempenho do algoritmo *Q-learning* com o modelo é geralmente baixo. Isso acontece porque o valor do reforço de um estado *s* pode apresentar muitos ruídos (*i.e.*, valor de reforço que pode ser satisfatório somente para a política de um agente). Geralmente, os ruídos são gerados pela política de ação parcial, acumulando reforços que não são satisfatórios, produzindo uma convergência irregular (mínimo local). É possível observar na figura 5.10 uma quantidade considerável de ruídos no início do processo de aprendizagem do algoritmo *Q-learning* com o modelo (região R3), especialmente quando há cooperação entre muitos agentes. No entanto, a ∂ (s,a) converge para Q*, mesmo compartilhando recompensas não satisfatórias. Em problemas de decisão sequencial um agente interage repetidamente no ambiente e tenta otimizar seu desempenho baseado nas recompensas recebidas. Assim, é difícil determinar as melhores ações em cada situação,

pois uma decisão específica pode ter um efeito prolongado, em função da influência sobre ações futuras.

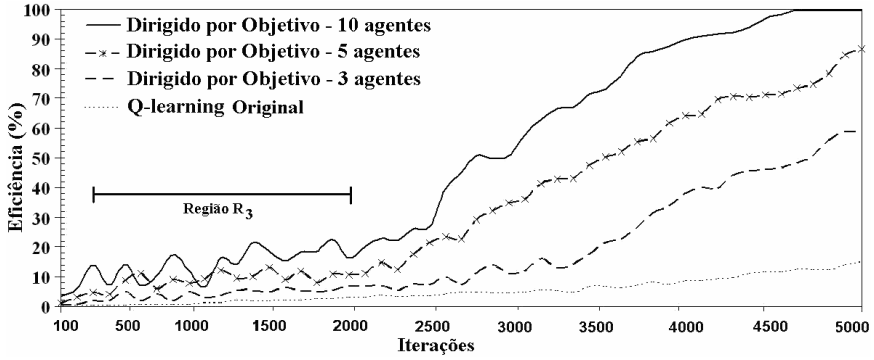


Figura 5.10: Modelo dirigido por objetivo

5.1.4 Modelo Híbrido de Aprendizagem

Em aprendizagem por reforço baseada em recompensas compartilhadas a descoberta de políticas não satisfatórias pode ocorrer em um dado momento, pois a troca de conhecimento entre os agentes pode gerar novas políticas intermediárias incompatíveis com uma rápida convergência. Como há mudanças no aprendizado de cada agente, é necessário que todos os agentes estejam atualizando e trocando suas recompensas enquanto interagem.

Não há garantia de convergência da política de ação baseada em recompensas compartilhadas usando os modelos descritos anteriormente. Observou-se empiricamente que políticas com estados e valores de recompensas inadequados, podem sofrer modificações com recompensas informadas por outras políticas parciais, melhorando a $\partial(s,a)$. No entanto, pode ocorrer o efeito contrário, onde políticas com estados com altas recompensas, podem se tornar menos interessantes para a política corrente, pois estados que produziam acertos passam a produzir erros.

Para resolver esse problema, foi desenvolvido um modelo híbrido de aprendizagem. Esse modelo surgiu a partir dos modelos discreto, contínuo e dirigido por objetivo e de constatações observadas nos experimentos. Pode-se notar que o comportamento da $\partial(s,a)$ com os modelos se altera em função da quantidade de iterações, de episódios e da quantidade de agentes. As figuras 5.11, 5.12 e 5.13 ilustram os modelos em ambientes com 400 estados.

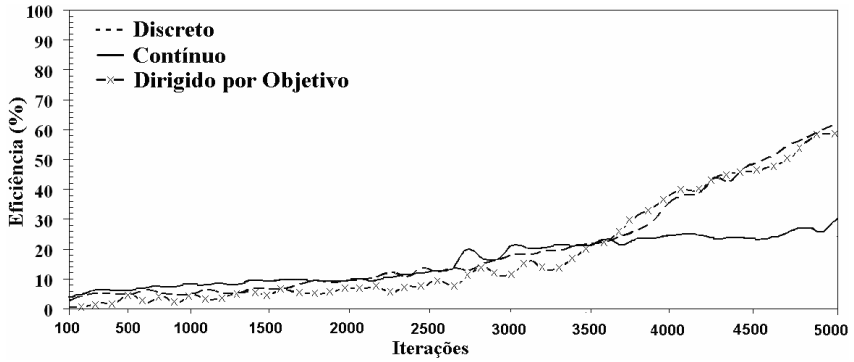


Figura 5.11: Ambiente 400 estados, 3 agentes

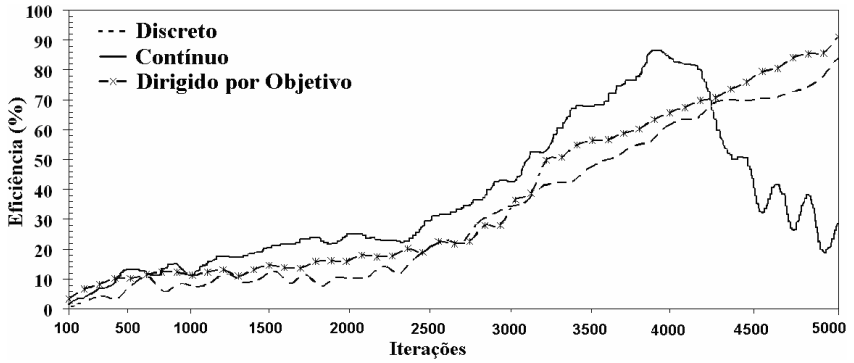


Figura 5.12: Ambiente 400 estados, 5 agentes

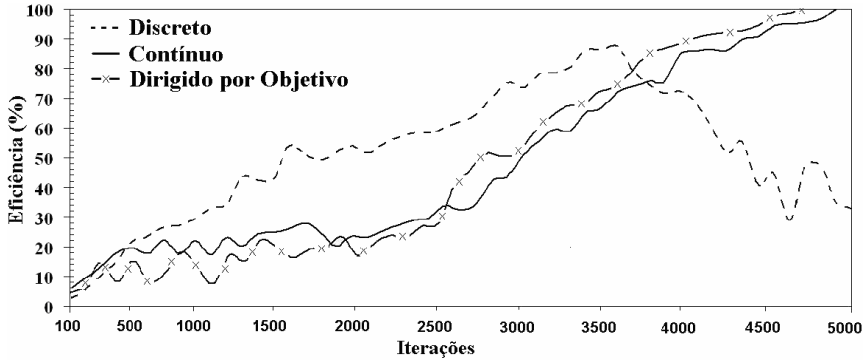


Figura 5.13: Ambiente 400 estados, 10 agentes

O modelo híbrido de aprendizagem utiliza as particularidades de cada modelo, permitindo a utilização das melhores características. Esse modelo descobre novas políticas de ação sem causar atrasos na aprendizagem, reduzindo possíveis conflitos entre ações com recompensas de políticas diferentes, melhorando a convergência. O modelo híbrido de aprendizagem funciona da seguinte forma: a cada iteração do algoritmo *Q-learning* com um modelo, o desempenho do agente com os modelos é comparado, gerando uma nova tabela de aprendizagem, de nome MH-$\partial(s,a)$ (Modelo Híbrido). Quando a condição de atualização do modelo é

alcançada, o agente inicia o aprendizado utilizando o modelo de melhor desempenho e a aprendizagem é transferida para a MH-$\partial(s,a)$. Portanto, a MH-$\partial(s,a)$ terá as melhores recompensas adquiridas dos modelos discreto, contínuo e dirigido por objetivo. Os resultados do modelo híbrido são apresentados a seguir.

5.1.5 Modelo Híbrido *vs.* Modelos Contínuo, Discreto e Dirigido por Objetivo

Nesta subseção são apresentados os principais resultados comparando o modelo híbrido de aprendizagem com os modelos discutidos anteriormente. Os parâmetros utilizados no modelo híbrido são os mesmos dos demais modelos. Os experimentos foram realizados em ambientes que variam entre 100 (10×10) e 400 (20×20) estados. Os resultados apresentados nas figuras 5.14 a 5.22 comparam o modelo híbrido e os demais modelos com diferentes quantidades de agentes e estados.

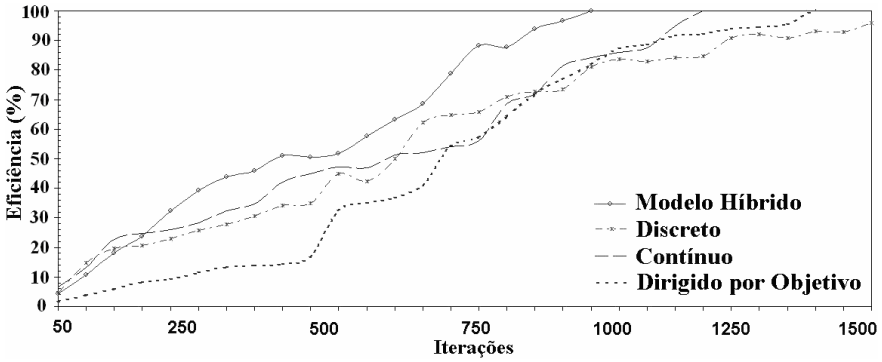


Figura 5.14: Ambiente de 100 estados; 3 agentes

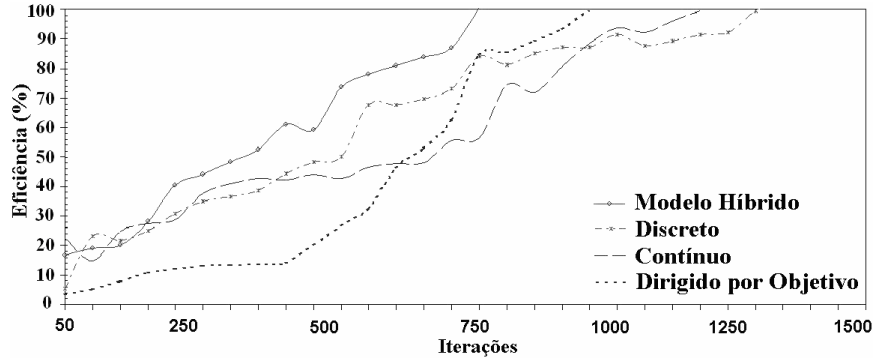


Figura 5.15: Ambiente de 100 estados; 5 agentes

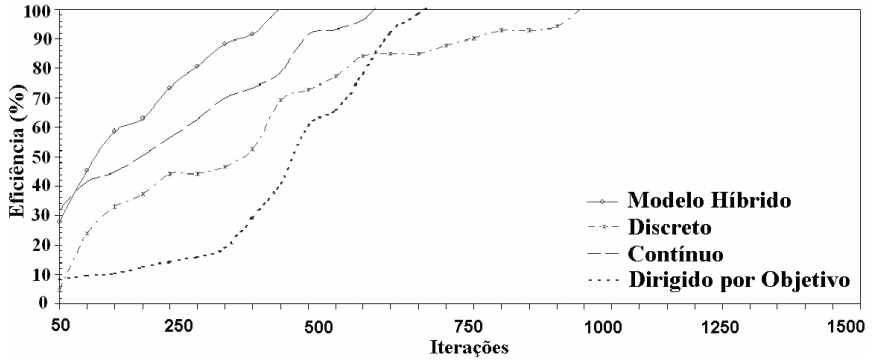


Figura 5.16: Ambiente de 100 estados; 10 agentes

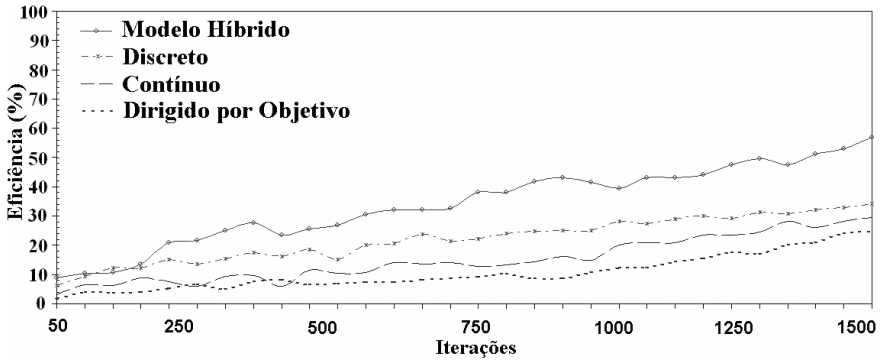


Figura 5.17: Ambiente de 250 estados; 3 agentes

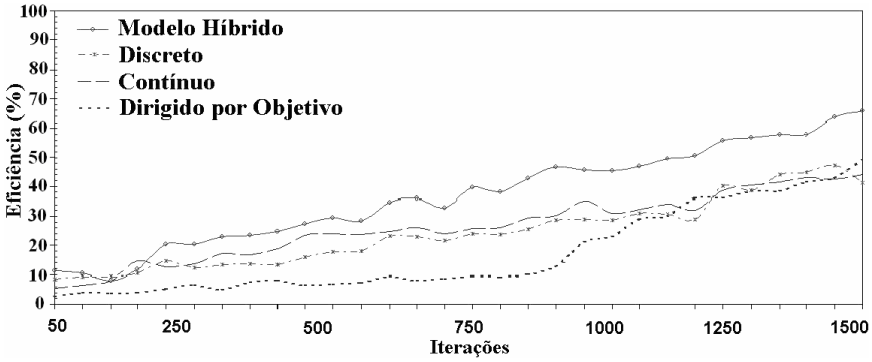


Figura 5.18: Ambiente de 250 estados; 5 agentes

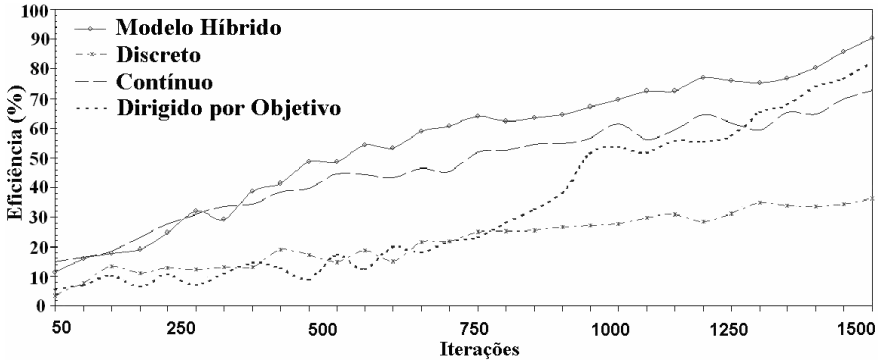


Figura 5.19: Ambiente de 250 estados; 10 agentes

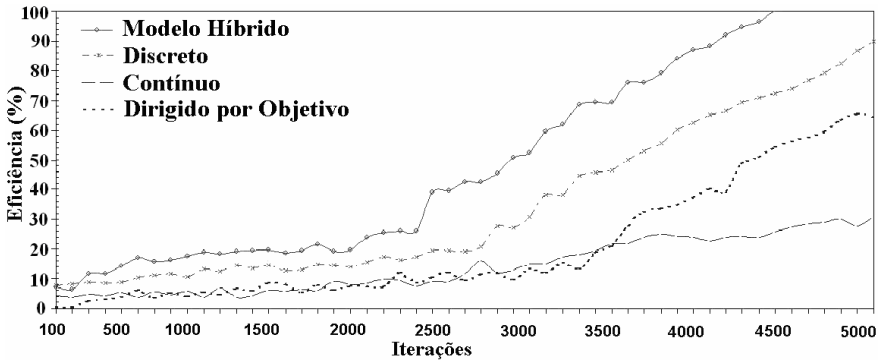


Figura 5.20: Ambiente de 400 estados; 3 agentes

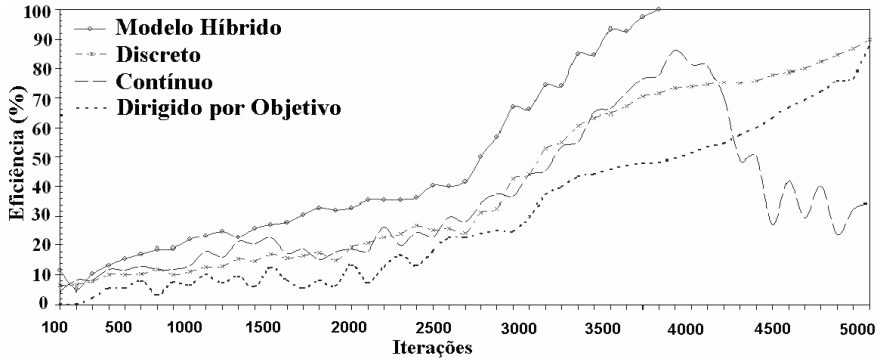


Figura 5.21: Ambiente de 400 estados; 5 agentes

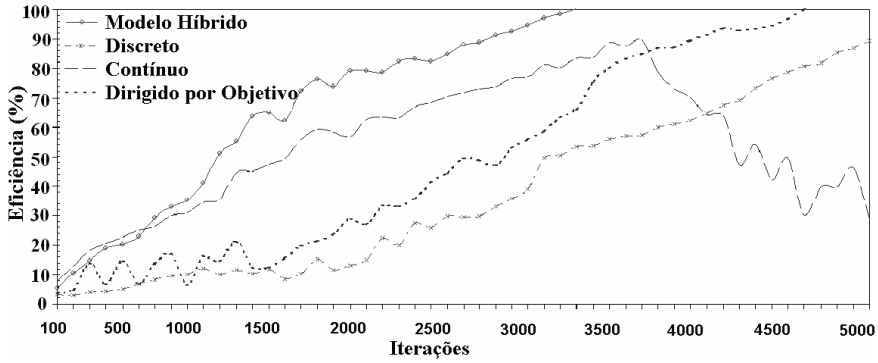


Figura 5.22: Ambiente de 400 estados; 10 agentes

As figuras 5.14 a 5.22 mostram que o modelo híbrido apresenta eficiência superior em relação às políticas geradas com os modelos discreto, contínuo e dirigido por objetivo. Geralmente, o modelo híbrido obtém desempenho superior em qualquer fase das iterações. Isso fez com que diminuísse significativamente o número de iterações necessárias para encontrar uma boa política de ação. Em ambientes com 100 estados o número de iterações diminuiu aproximadamente 23,4% quando utilizados 3 agentes; 27,1% com 5 agentes e 40,4% com 10 agentes. Nos ambientes com 250 estados o modelo híbrido consegue diminuir o número de iterações em 20% com 3 agentes; 22,8% com 5 agentes e 33,9% com 10 agentes. Já nos ambientes com 400 estados o número de iterações diminuiu 18,6% com 3 agentes; 23,3% com 5 agentes e 28,7% com 10 agentes. A tabela 5.1 sumariza a superioridade média da eficiência do modelo híbrido comparado com o melhor dentre os demais modelos.

Tabela 5.1: Superioridade média do modelo híbrido em relação aos modelos discreto, contínuo e dirigido por objetivo

	Quantidade de agentes		
#Estados	3	5	10
100	17,6%	69,8%	37,2%
250	50,3%	31,4%	33,7%
400	45,1%	40,9%	37,6%

É possível observar que o desempenho dos agentes com o modelo híbrido é melhor do que o desempenho das políticas de ação baseadas em recompensas individuais. O bom desempenho dos agentes que cooperam utilizando o modelo híbrido é decorrente de $\partial(s,a)$ gerada a partir de valores de aprendizagem descobertos de forma colaborativa. Assim, os agentes conseguem gerar políticas com boas recompensas para aproximar os valores aprendidos de uma

boa política. Ademais, o número de iterações diminuiu significativamente com o modelo híbrido.

5.2 Análise do *Ant-Q*

No capítulo 4 foram discutidas algumas técnicas para problemas de otimização combinatória a partir de algoritmos de colônia de formigas, enquanto na seção anterior observou-se como a coordenação dos agentes pode ser melhorada a partir de interações com as recompensas sociais geradas por algoritmos de aprendizagem por reforço. A aprendizagem por reforço e o algoritmo *Ant-Q* se relacionam devido à maneira que este algoritmo estabelece o comportamento dos agentes com as recompensas geradas pelas interações e sua capacidade de combinar técnicas de aprendizagem por reforço com buscas heurísticas para melhorar a exploração. Antes de discutirmos detalhadamente a relação dos algoritmos de colônia de formigas com as redes sociais, é apresentado um *framework* de teste desenvolvido para demonstrar o desempenho dos agentes com o algoritmo *Ant-Q* e para descrever o comportamento do *Ant-Q* com diferentes cenários e parâmetros.

O *framework* é capaz de mostrar o impacto da variação dos parâmetros e da quantidade de agentes no algoritmo. O algoritmo foi testado no problema do caixeiro viajante, sabidamente *NP*-completo. O problema do caixeiro viajante é um problema clássico de otimização combinatória, frequentemente utilizado na computação para demonstrar problemas de difícil resolução, podendo ser formalizado pela teoria dos grafos. O objetivo é encontrar o percurso (caminho) de menor distância, passando por todos os estados uma única vez até a origem (ciclo *hamiltoniano* de menor custo).

Uma alternativa para solucionar o problema do caixeiro viajante é testar as permutações possíveis, empregando algoritmos de busca exaustiva para encontrar o percurso com menor custo. No entanto, dado que a quantidade de permutações é $(n - 1)!$, tal alternativa torna-se impraticável para a solução na maioria das vezes. Portanto, diferentemente das técnicas exaustivas, algoritmos heurísticos como o *Ant-Q*, buscam soluções desejáveis em menor tempo. Mesmo sem garantir a melhor solução (política ótima), o ganho computacional favorece a solução.

As estruturas internas do *framework* são formalizadas com os algoritmos das figuras 5.24 a 5.28 que compõem o algoritmo *Ant-Q* apresentado na figura 5.23.

```
Algoritmo Ant-Q
01  Início
02   Distribua os estados no plano cartesiano
03   Calcule e distribua o AQ0, conforme a equação 5.1
04     Para (cada episódio) Repita:
05      Defina a posição inicial dos Agentes
06       Enquanto (existirem estados a serem visitados) Faça
07         Para (cada Agente) Repita:
08          Se (q(rand(0..1) <= q0) Então
09           Escolha a ação conforme a equação 4.21
10          Senão
11           Escolha a ação conforme a equação 4.22
12      Fimse
13      Atualize o feromônio da aresta i com a equação
        4.22
14    Fimpara
15  Fimenquanto
16       Compute o melhor custo do episódio
17       Atualização global, conforme a equação 5.3
18     Fimpara
19  Fim
```

Figura 5.23: Pseudocódigo do *Ant-Q* (baseado em Gambardella e Dorigo, 1995)

O algoritmo da figura 5.23 é descrito da seguinte maneira. Inicialmente, é calculado com a equação 5.1 o valor inicial do feromônio (AQ_0):

$$\frac{1}{avg \times n} \tag{5.1}$$

$$d_{su} = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2} \tag{5.2}$$

onde *avg* é a média das distâncias euclidianas dos estados pares *su* calculada pela equação 5.2, e *n* é o número de agentes no sistema.

```
Função calculaAQ0(numEstados, numAgentes)
01  Início
02    Para (cada par de estados) Repita:
03      Soma ← calcDist(xs,ys,xu,yu); //conforme a equação 5.2
04    Fimpara
05   Media ← soma / numEstados;
06   AQ0 ← 1 / (media * numAgentes);
07   return (AQ0)
08  Fim
```

Figura 5.24: Cálculo para AQ_0

O algoritmo da figura 5.24 é usado para calcular o valor de AQ_0, que será atribuído a todas as arestas que compõem o grafo. Assim, os agentes podem selecionar os estados baseados no valor do feromônio ou da heurística (proporcional ao inverso de sua distância).

Um parâmetro importante no algoritmo *Ant-Q* é o q_0, que define o tipo de exploração adotada pelo agente a cada ação. O valor é gerado de maneira aleatória no intervalo [0,1]. Caso o valor gerado seja inferior ou igual a q_0, o agente adota a ação do tipo *gulosa*, *i.e.* escolhe a aresta com maior recompensa (`max`), figura 5.25; caso contrário, o agente utiliza a estratégia exploratória (figura 5.26).

```
Função exploitation()
01  Início
02    Para (cada aresta s,i) Repita:
03      Se (i >= max) Então
04        max ← i
05      Fimse
06    Fimpara
07    Return (max)
08  Fim
```

Figura 5.25: Função *exploitation*

```
Função exploration()
 01  Início
 02    Para (cada estado a ser visitado) Repita:
 03      probabilidade ← calcProb();//conforme a equação 4.22
 04    Fimpara
 05    Para (cada estado a ser visitado) Repita:
 06      Se (probabilidade <= rand(estado)) Então
 07        estadoSelecionado = estado
 08      Fimse
 09    Fimpara
 10   Return (estadoSelecionado)
 11  Fim
```

Figura 5.26: Função *exploration*

Para cada ação do agente, o valor na aresta dos estados adjacentes é atualizado conforme a equação 5.3.

$$AQ(s,u) \leftarrow (1-\alpha).AQ(s,u) + \alpha.\left(\Delta AQ(s,u) + \gamma.\max_{i \in N_j^k(t)} AQ(j,i)\right) \qquad (5.3)$$

À direita da figura 5.24, estão posicionados os parâmetros do algoritmo e do ambiente, onde δ e β são respectivamente os parâmetros da regra de transição, e y e α são os parâmetros de aprendizagem do algoritmo. As variáveis m_k, S e n_t representam o número de agentes, a quantidade de estados e o número de episódios respectivamente, onde n_t é usado como critério de parada do algoritmo.

O valor para `max` é calculado investigando todas as arestas adjacentes (estados possíveis a *i*). O maior valor encontrado é usado na solução.

Um importante aspecto do algoritmo é a forma de atualização na tabela de aprendizagem, podendo ocorrer de maneira global ou local. A atualização global ocorre no final de cada episódio, onde é escolhida a política de menor custo e atualizados os valores dos estados com o parâmetro de reforço (figura 5.27). Esse procedimento é similar ao modelo dirigido por objetivo descrito na seção 5.1. A equação 5.4 é usada para calcular o valor de $\Delta AQ(s,u)$, que será o reforço da atualização global (figura 5.28).

$$\Delta AQ(s,u) = \begin{cases} W \\ L_{Best} \end{cases} \quad (5.4)$$

onde W é uma variável parametrizada com o valor 10 e L_{best} é custo total do percurso. Já a atualização local ocorre a cada ação dentro do episódio, onde $\Delta AQ(s,u)$ terá valor zero.

```
Função atualizaçãoLocal(s,u)
 01  Início
 02   max ← calcmax()
 03   AQ(s,u) = calcFeromon(s,u,max);//conforme a equação 5.3
 04  Fim
```

Figura 5.27: Atualização local

```
Função atualizaçãoGlobal(melhorRota)
 01  Início
 02  Reforço ← calcReforço(); //conforme a equação 5.4
 03   Para (cada aresta pertencente a melhor política) Repi-
     ta:
 04    atualizaçãoLocal(Reforço)
 05   Fimpara
 06  Fim
```

Figura 5.28: Atualização global

5.2.1 Resultados Experimentais

Experimentos são mostrados para avaliar o impacto dos parâmetros de aprendizagem do algoritmo *Ant-Q*. Os parâmetros de aprendizagem podem influenciar a coordenação dos agentes durante a interação, e, se ajustados inadequadamente, podem ocasionar atrasos no aprendizado ou até mesmo causar situações inesperadas de transição, convergindo para uma solução não satisfatória. Portanto, os experimentos realizados com o algoritmo avaliam sua eficiência considerando fatores como: variações na taxa de aprendizagem, fator de desconto, taxa de exploração, regras de transição e quantidade de agentes no sistema.

Para analisar a eficiência dos parâmetros do algoritmo, foram gerados 5 cenários diferentes para cada tipo de experimento, em ambientes de 35, 45 e 55 estados (figura 5.29). O aprendizado em cada cenário foi realizado 15 vezes pelo algoritmo (15 amostras), pois se observa que fazendo experimentos em um mesmo ambiente, com entradas iguais, podem ocorrer variações na eficiência gerada pelo algoritmo. Isto ocorre porque as ações dos agentes são probabilísticas e os valores gerados durante sua aprendizagem são estocásticos. Portanto, as políticas de ação dos agentes podem variar de um experimento para outro. Assim, a eficiência

apresentada nesta seção representa a média de todos os experimentos gerados nos 5 cenários com 15 amostras em cada ambiente. Esse número de repetições foi suficiente para avaliar a eficiência do algoritmo, pois observamos que a partir deste número os resultados dos experimentos não alteravam significativamente a qualidade das políticas. O eixo Y dos gráficos das figuras 5.31 a 5.35 apresenta o custo da política em % encontrada com cada parâmetro. O eixo X do gráfico indica o valor do parâmetro. Em muitos problemas de otimização não é possível *a priori* conhecer a política ótima. Para calcular a eficiência do algoritmo em percentual (eixo Y), é usada uma escala para cada política, onde 100% indica a política de menor custo e 0 caso contrário.

A quantidade de agentes no ambiente é igual à quantidade de estados. Inicialmente, os parâmetros foram configurados com os seguintes valores: δ= 1; β= 2; γ= 0,3; α= 0,1; q_0= 0,9 e W= 10. Foi utilizada como critério de parada a quantidade de 300 episódios. Cabe observar que dependendo do tamanho e da complexidade do ambiente, esse número não é suficiente para encontrar a melhor política. No entanto, o objetivo dos experimentos é avaliar o impacto dos parâmetros na convergência dos agentes e não a qualidade da política encontrada.

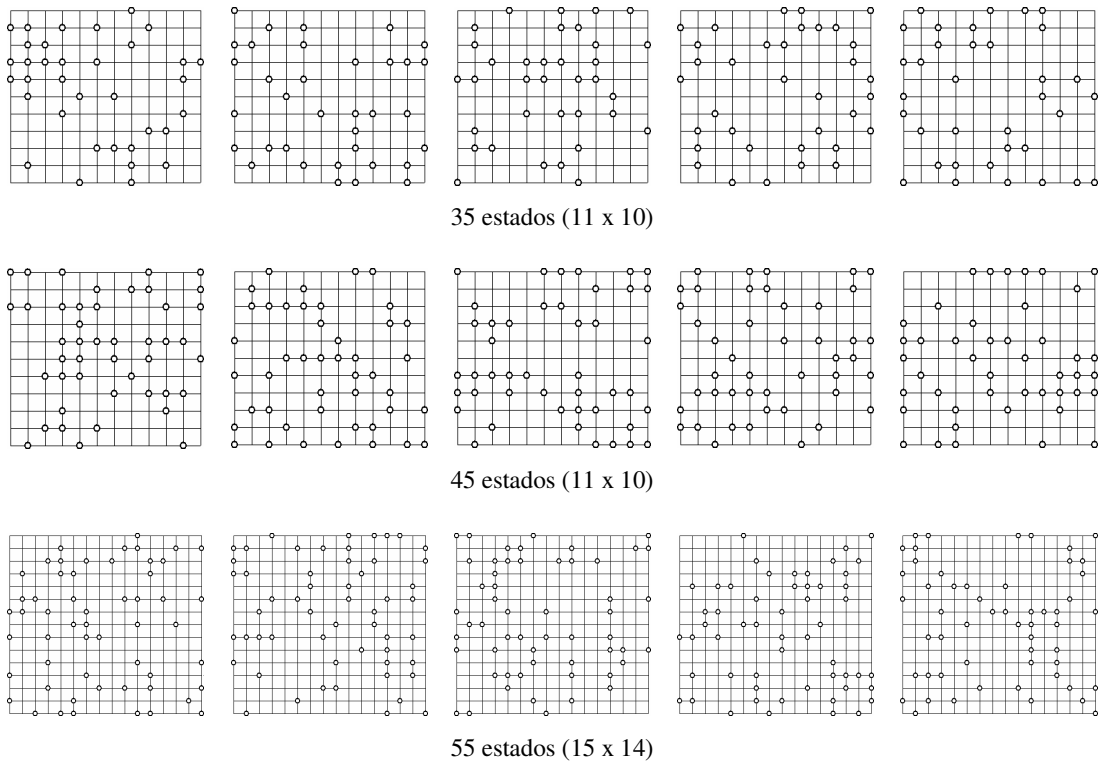


Figura 5.29: Ambientes usados na simulação, onde os estados estão expressos em um sistema euclidiano de coordenadas 2D

Antes de discutirmos os resultados variando os fatores de aprendizagem, é importante lembrar que observamos a necessidade de poucos episódios para encontrar as melhores políticas. Isso acontece devido à influência da heurística, que foi parametrizada com o dobro do valor da influência do feromônio. As imagens da figura 5.30 ilustram a evolução da política em um ambiente com 55 estados a cada 50 episódios.

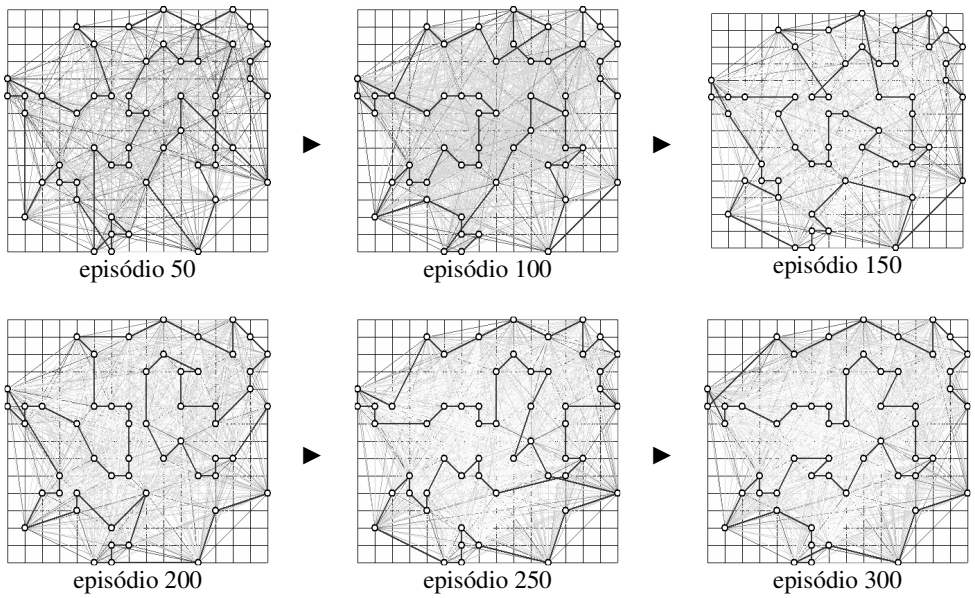


Figura 5.30: Evolução da política a cada 50 episódios

5.2.1.1 Taxa de Aprendizagem

A taxa de aprendizagem α indica a importância do feromônio computado ao estado selecionado. Para verificar os melhores valores para α foram realizados experimentos nos três ambientes usando valores entre 0 e 1. Os melhores valores para α estão entre 0,1 e 0,2. Valores superiores fazem com que, os agentes ao estabelecerem uma melhor ação em um determinado estado do ambiente, não efetuassem outras ações na busca de caminhos de menor custo. Valores inferiores não dão a devida importância ao aprendizado, não permitindo que os agentes selecionem caminhos diferentes da política corrente. O melhor valor de α para a política foi de 0,1, sendo usado nos demais experimentos. Observamos ainda que, quanto menor a taxa de aprendizagem, menor é a variação da política. A figura 5.31 apresenta a eficiência das taxas de aprendizagem no intervalo [0,.,1].

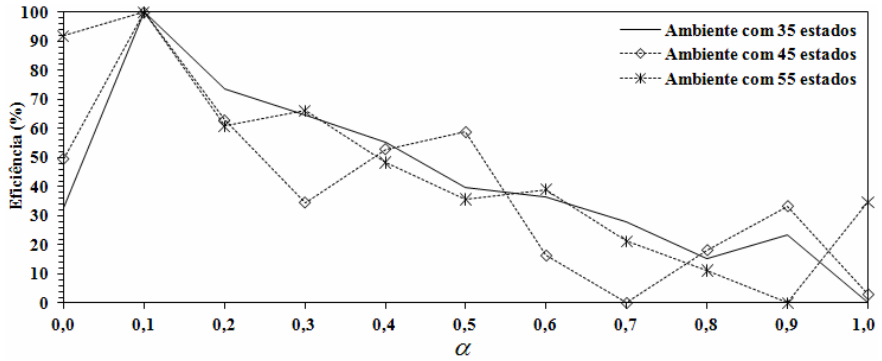


Figura 5.31: Eficiência da taxa de aprendizagem

5.2.1.2 Fator de Desconto

O fator de desconto determina o peso temporal relativo dos reforços recebidos. Os melhores valores para o fator de desconto estão entre 0,2 e 0,3 conforme apresentado na figura 5.32. Valores diferentes de 0,2 e 0,3 mostraram-se ineficientes para a convergência, tendo pouca relevância para a aprendizagem dos agentes. Quando o valor é superior a 0,3, ele apresenta relevância excessiva, induzindo os agentes a ótimos locais.

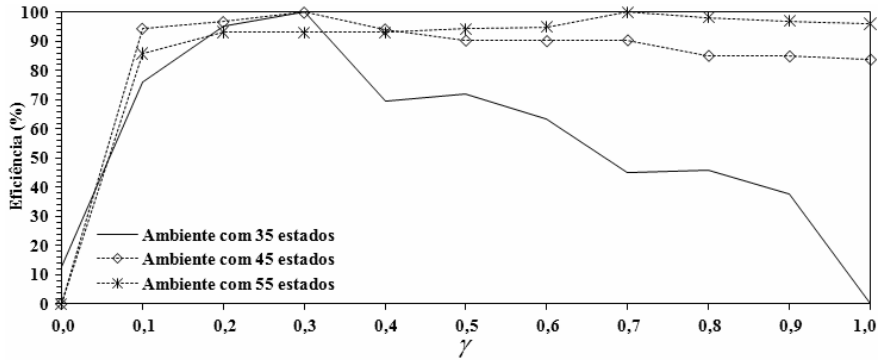


Figura 5.32: Eficiência do fator de desconto

5.2.1.3 Taxa de Exploração

A taxa de exploração q_0 indica a probabilidade de um agente escolher um determinado estado. Os experimentos para encontrar a melhor taxa de exploração foram realizados em ambientes com estados e tamanhos diferentes. Os melhores valores utilizados estão entre 0,8 e 1. À medida que o valor se aproxima de zero, as ações dos agentes vão se tornando cada vez mais aleatórias, consequentemente as soluções começam a não ser satisfatórias.

O melhor valor encontrado para q_0 é 0,9. Com isso, agentes selecionam os caminhos de menor custo e com maior concentração de feromônio. Com q_0= 0,9 a busca é praticamente

gulosa, pois 0,1 será a probabilidade de escolher os demais caminhos. A figura 5.33 mostra os resultados para q_0 no intervalo [0,.,1].

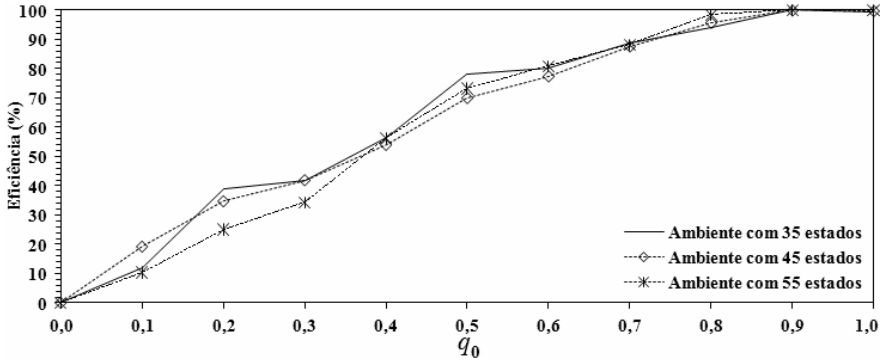


Figura 5.33: Resultados do parâmetro de exploração

5.2.1.4 Regra de Transição

Os experimentos alterando os fatores δ e β foram realizados em ambientes com estados e tamanhos diferentes. Conforme observado, o algoritmo é dependente de heurísticas, onde o peso é representado pelo parâmetro β. Para obter bons resultados, o valor de β deve ser pelo menos 65% do valor de δ. A figura 5.34 ilustra os resultados variando os fatores δ e β.

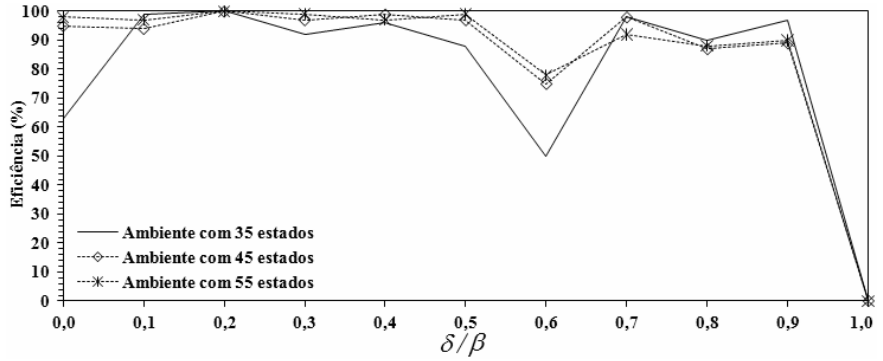


Figura 5.34: Resultados da regra de transição (δ e β)

5.2.1.5 Quantidade de Agentes

Para avaliar o impacto da quantidade de agentes no sistema, foram utilizados de 10 a 80 agentes. A figura 5.35 mostra que as melhores políticas são encontradas quando a quantidade de estados é igual à quantidade de agentes no sistema ($m_k = x$), onde x é a quantidade de estados e x_i é a variação dos agentes no sistema. Pode-se observar que a quantidade superior de agentes ao número de estados ($m_k > x$) mostra-se inadequada para boas soluções, apresentando comportamento de estagnação. Assim, ao encontrarem uma solução, agentes evitam a

busca por outros caminhos, determinando um máximo local. Quando a quantidade de agentes é inferior ao número de estados ($m_k < x$), o número de episódios teve que ser aumentado de maneira exponencial para encontrar as melhores soluções.

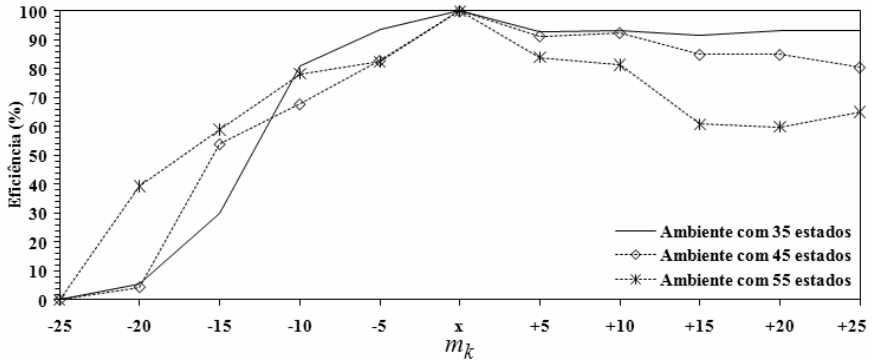


Figura 5.35: Quantidade de agentes (m_k)

5.2.2 Estratégias de Atualização de Políticas para Ambientes Dinâmicos

Observou-se nas seções anteriores que algoritmos de colônia de formigas são eficientes quando os parâmetros de aprendizagem são ajustados e quando não ocorrem alterações no ambiente que modificam a política ótima. No entanto, em ambientes dinâmicos não se tem a garantia da convergência do algoritmo *Ant-Q*, pois sabe-se que esse algoritmo foi originalmente desenvolvido e aplicado em problemas estáticos, onde a função objetivo não se altera no tempo. No entanto, raramente há problemas do mundo real que são estáticos, devido a mudanças de prioridades por recursos, alterações nos objetivos e tarefas que não são mais necessárias. Essas particularidades caracterizam um ambiente dinâmico.

Conforme discutido na subseção 4.4.7, várias técnicas baseadas em algoritmos de colônia de formigas foram desenvolvidas para melhorar a habilidade de exploração dos algoritmos em ambientes dinâmicos. Essas técnicas podem ser usadas como alternativas para melhorar a solução assim que o ambiente é alterado. As abordagens propostas são baseadas em técnicas que usam estratégias para melhorar a exploração usando a transição probabilística do *ant colony system*, aumentando a exploração do espaço de estados. Dessa forma, a decisão de transição mais aleatória é usada, variando alguns parâmetros onde a nova informação heurística influencia a seleção das arestas mais desejáveis.

Alguns trabalhos utilizam regras de atualização nas arestas da solução, incluindo um componente de evaporação similar à regra de atualização do *ant colony system*. Dessa forma, ao longo do tempo a concentração do feromônio diminui, fazendo que os estados menos favoráveis sejam menos explorados nos episódios futuros. Para isso, uma alternativa seria reinicia-

lizar o valor do feromônio após observar as alterações no ambiente, mantendo uma referência para as melhores soluções encontradas. Se identificado o local da alteração no ambiente, o feromônio dos estados adjacentes é reinicializado, fazendo com que os estados se tornem mais desejados. Se um estado não é satisfatório, reforços podem ser menores (geralmente proporcional a qualidade da solução), e ao longo do tempo, tornam-se menos desejáveis devido à redução do feromônio pela evaporação.

É possível observar que a maioria dos trabalhos propostos concentra seus esforços em melhorar as regras de transição empregando estratégias sofisticadas para a convergência. No entanto, os experimentos mostram que tais métodos não conseguem bons resultados em ambientes altamente dinâmicos e onde o tamanho do espaço de busca é incerto.

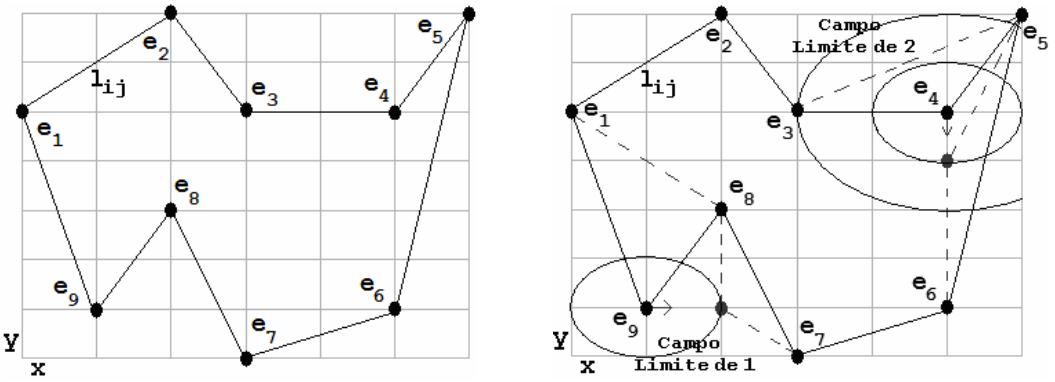
Para isso, são apresentadas nesta seção algumas estratégias que foram desenvolvidas para a atualização de políticas geradas por recompensas (feromônios) para ambientes dinâmicos. É verificado que quando os parâmetros de algoritmos baseados em recompensas são ajustados inadequadamente pode ocorrer atrasos no aprendizado e convergência para uma solução não-satisfatória. Além disso, esse problema é agravado em ambientes altamente dinâmicos, pois o ajuste dos parâmetros de tais algoritmos não é suficiente para garantir convergência.

As estratégias desenvolvidas modificam valores de feromônio, melhorando a coordenação entre os agentes e permitindo convergência mesmo quando há mudanças na posição cartesiana dos estados do ambiente. O objetivo das estratégias é encontrar o equilíbrio ótimo da recomposição da política, que permita explorar novas soluções usando informações de políticas passadas. Equilibrar o valor do feromônio equivale a reajustar as informações das ligações, dando ao processo de busca flexibilidade para encontrar uma nova solução quando o ambiente é alterado, compensando a influência das políticas passadas na construção de novas soluções.

Uma das estratégias de atualização desenvolvidas é inspirada nas abordagens propostas em (Guntsch e Middendorf, 2001) e (Lee *et al.* 2001b), reinicializando localmente os valores de feromônio quando alterações no ambiente são identificadas. Este método é chamado de estratégia *média global*. Essa estratégia atribui às ligações adjacentes dos estados alterados, a média de todos os valores de feromônio da melhor política. A estratégia média global é limitada, pois não observa a intensidade de alteração do ambiente. Por exemplo: muitas vezes, boas soluções com estados alterados podem diminuir a qualidade da solução, sendo necessário atualizar apenas parte da política de ação. A estratégia *distância global* atualiza o feromônio dos estados considerando a distância euclidiana entre todos os estados do ambiente com a distância euclidiana do ambiente alterado. Se o custo da política aumenta com a alteração, então

o valor do feromônio diminui proporcionalmente, caso contrário, o valor é aumentado. A estratégia *distância local* é similar à estratégia *distância global*, no entanto, a atualização do feromônio é proporcional à diferença na distância euclidiana dos estados que foram alterados.

Antes de discutirmos com mais detalhes como as estratégias atribuem valores para a política corrente, é apresentado como as alterações ocorrem no ambiente. Os estados do ambiente podem ser alterados devido a fatores como, escassez dos recursos, mudança de objetivos ou atribuições de tarefas, de tal maneira que estados podem ser inseridos, excluídos, ou simplesmente movimentados no ambiente. Tais características podem ser encontradas em diferentes aplicações como gerenciamento de tráfego, redes de sensores, gerenciamento de cadeias de suprimentos ou redes de comunicação móveis.



(a) Posição dos estados antes da alteração (A) (b) Posição dos estados após alteração (A')

Figura 5.36: Dinâmica do ambiente

A figura 5.36 mostra uma representação simplificada de um ambiente com 9 estados. A figura 5.36a ilustra o ambiente antes da alteração, enquanto a figura 5.36b ilustra o ambiente depois das alterações. A configuração do ambiente é mostrada na tabela 5.2.

Tabela 5.2: Estados antes e após as alterações

Antes das alterações (A)		**Após alterações (A')**	
estados	ligações	estados	ligações
$e_1(0,5)$	1→2, 1→9	$e_1(0,5)$	1→2, **1→8**
$e_2(2,7)$	2→3, 2→1	$e_2(2,7)$	2→3, 2→1
$e_3(3,5)$	3→2, 3→4	$e_3(3,5)$	**3→5**, 3→2
$e_4(5,5)$	4→3, 4→5	**$e_4(5,4)$**	**4→6**, 4→5
$e_5(6,7)$	5→4, 5→6	$e_5(6,7)$	5→4, 5→3
$e_6(5,1)$	6→5, 6→7	$e_6(5,1)$	**6→4**, 6→7
$e_7(3,0)$	7→6, 7→8	$e_7(3,0)$	**7→9**, 7→6
$e_8(2,3)$	8→9, 8→7	$e_8(2,3)$	8→9, **8→1**
$e_9(1,1)$	9→1, 9→8	**$e_9(2,1)$**	**9→7**, 9→8

Pode-se observar que a alteração dos estados e_4 e e_9 irá atribuir à política corrente seis novas ligações. As alterações no ambiente são realizadas de maneira aleatória, considerando alterações na posição cartesiana dos estados mas restringindo-se ao tamanho do campo limite, ou seja, adjacentes à uma posição cartesiana.

Portanto, uma mudança introduzida no ambiente pode modificar a localização (posição) de um estado e isso pode causar diferenças parciais entre a política corrente e a política ótima, causando temporariamente políticas indesejadas e erros. As estratégias devem atualizar o valor do feromônio de cada ligação dos estados alterados, conforme as características de cada estratégia.

A. Estratégia Média Global

A estratégia média global não considera a intensidade da alteração no ambiente, no entanto consegue perceber os estados alterados. É atribuído às ligações incidentes aos estados alterados o valor médio do feromônio de todas as ligações da melhor política corrente (Q). Diferentemente de outros trabalhos, que reinicializam o feromônio sem considerar o valor aprendido, a estratégia média global reutiliza os valores de políticas passadas para estimar os valores de atualização. A equação 5.5 mostra como são computados os valores desta estratégia:

$$média_global = \frac{\sum_{l \in Q} AQ(l)}{n_l} \tag{5.5}$$

onde n_l é o número de ligações e AQ(l) é o valor do feromônio da ligação l.

B. Estratégia Distância Global

Na estratégia distância global é calculada a distância entre todos os estados e o resultado é comparado com a distância dos estados do ambiente alterado. Assim, esta estratégia considera a intensidade total de alteração no ambiente. Se a distância entre os estados aumenta, então a atualização do valor do feromônio é inversamente proporcional em relação à distância. Caso o custo da distância entre os estados diminua, então o valor é aumentado na mesma proporção. A equação 5.6 é usada para estimar o valor de atualização nas ligações dos estados do ambiente A'.

$$distância_global = \frac{\sum_{s=1}^{n_e}\sum_{u=s+1}^{n_e} d_A(l_{su})}{\sum_{s=1}^{n_e}\sum_{u=s+1}^{n_e} d_{A'}(l_{su})} \times AQ(l_{su}) \tag{5.6}$$

onde n_e é o número de estados, A' o ambiente após as alterações e d a distância euclidiana entre os estados.

C. Estratégia Distância Local

A estratégia distância local é similar à estratégia anterior, no entanto atualiza o feromônio somente nas ligações incidentes aos estados modificados. Dessa forma, cada ligação é atualizada proporcionalmente à distância dos estados adjacentes que modificaram, tornando a atualização dessa estratégia local, melhorando a convergência quando ocorrem poucas alterações no ambiente. A equação 5.7 é usada para computar o valor da atualização nas ligações:

$$distância_local = \frac{d_A(l_{su})}{d_{A'}(l_{su})} \times AQ(l_{su}) \tag{5.7}$$

O pseudocódigo da figura 5.23 foi modificado com a inclusão das estratégias supracitadas, produzindo o algoritmo da figura 5.37.

```
Algoritmo Ant-Q com estratégias ()
01  Início
02    Distribua os estados
03    Calcule o feromônio inicial com a equação 5.1 e o dis-
      Tribua nas ligações
04    Para cada episódio Repita:
05      Defina a posição inicial dos Agentes
06      Enquanto existirem estados a serem visitados Faça:
          // Nesse caso, lista tabu <> ϕ
07        Para cada Agente repita:
08          Se (q(rand(0..1) <= q0) Então
09            Escolha a ação conforme equação 4.21
10          Senão
11            Escolha a ação conforme equação 4.22
12          Fimse
13          Atualize o feromônio da ligação (s,u) usando a
            atualização local
14        Fimpara
15      Fimenquanto
16      Calcule o custo da melhor política do episódio
17      Realize a atualização global, usando as equações 5.3
        e 5.4
18      Se ocorrer alterações no ambiente Então
19        Caso: (Estratégia média_global) Então
20          valor ← estrategia1(); // equação 5.5
21        Caso: (Estratégia distância_global) Então
22          valor ← estrategia2(); // equação 5.6
23        Caso: (Estratégia distância_local) Então
24          valor ← estrategia3(); // equação 5.7
25        Para cada estado alterado Faça:
26          Para cada ligação (s,u) incidente ao estado alte-
            rado Faça:
27            AQ(s,u) ← valor;
28          Fimpara
29        Fimpara
30      Fimse
31    Fimpara
32  Fim
```

Figura 5.37: Pseudocódigo do *Ant-Q* com as estratégias (modificado de 5.26)

5.2.2.1 Resultados com as Estratégias de Atualização

Para avaliar as estratégias propostas na seção anterior, foram gerados ambientes dinâmicos com 35 estados. O comportamento dos agentes foi avaliado considerando a porcentagem de mudança gerada pelo ambiente a cada 100 episódios. Essa janela temporal foi utilizada porque em trabalhos anteriores foi observado que em ambientes de 35 estados ela permitiu ao algoritmo conquistar boa convergência (Ribeiro *et al.* 2009c).

A alteração ocorre da seguinte maneira: a cada 100 episódios, o ambiente produz um conjunto de alterações. As mudanças são realizadas aleatoriamente, de tal maneira que simule alterações em locais parcialmente conhecidos ou sujeitos a ruído. Dessa forma, ambientes com 35 estados terão 7 estados alterados quando 20% de mudança ocorrer. Ademais, foram simuladas alterações considerando o espaço do campo limite com profundidade 1 e 2, limitando assim a mudança da posição de um estado e permitindo simular dinâmicas graduais próximas de problemas do mundo real.

Os resultados dos experimentos comparam as três estratégias com a política descoberta com o *Ant-Q* original. Os parâmetros de aprendizagem utilizados na simulação são os mesmos empregados na seção 5.2. Cada estratégia permitiu que na maioria das vezes o número de episódios diminuísse, pois a combinação das recompensas pôde estimar valores melhores, que levaram os agentes a uma convergência quando a política é atualizada. As figuras 5.38, 5.39, 5.40 e 5.41 demonstram a convergência do algoritmo em ambientes com 35 estados. O eixo X dessas figuras indica os episódios. Quando o percurso de menor custo é encontrado, a eficiência é 100% (eixo Y).

Observando as figuras 5.38 à 5.41, é possível notar que a política global com as estratégias é superior a do *Ant-Q* original. A estratégia média global mostra-se mais adequada para ambientes com variações maiores (figuras 5.39 e 5.41). Isso ocorre porque a estratégia utiliza todos os valores de reforços do ambiente. No entanto, os agentes sofrem para convergir quando o ambiente tem poucas alterações, pois estados alterados terão recompensas menores que os estados que constituem a melhor solução atual. Já a estratégia distância global mostra-se mais robusta em ambientes com poucas variações (figuras 5.38 e 5.40). Quando o ambiente é alterado, a estratégia age nos estados atualizando a recompensa proporcionalmente à quantidade de alterações do ambiente. Dessa forma, o efeito da atualização diminui o impacto após as mudanças, fazendo que os agentes convergissem uniformemente. A estratégia distância local considera somente as alterações locais, dessa forma, a atualização da política com tal estratégia é melhor quando os valores dos reforços são maiores, ou seja, nos episódios finais.

De maneira geral, a política global das estratégias consegue acumular bons valores de reforços com um número pequeno de episódios de aprendizagem. As estratégias atualizam a política global acumulando bons valores de reforços, desde que haja uma quantidade de episódios necessária. Nos episódios iniciais da aprendizagem, a política é menos sensível às estratégias, o que melhora o desempenho da política após a atualização. Algumas estratégias podem estimar valores não adequados para a política, principalmente após muitos episódios e mudanças no ambiente, ocasionando máximos locais.

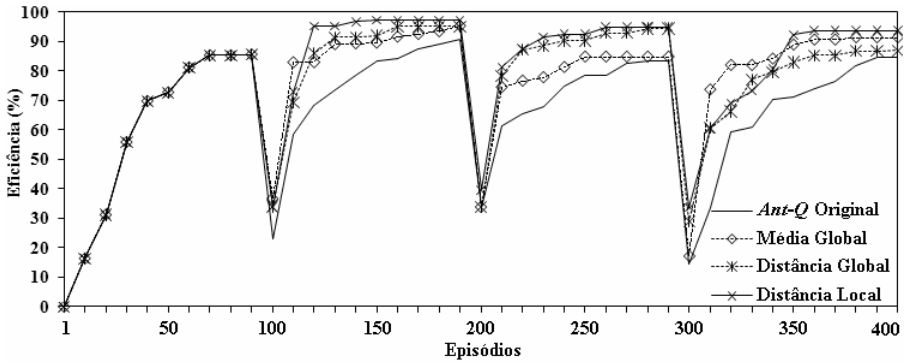


Figura 5.38: Campo limite de 1; 10% de alterações a cada 100 episódios

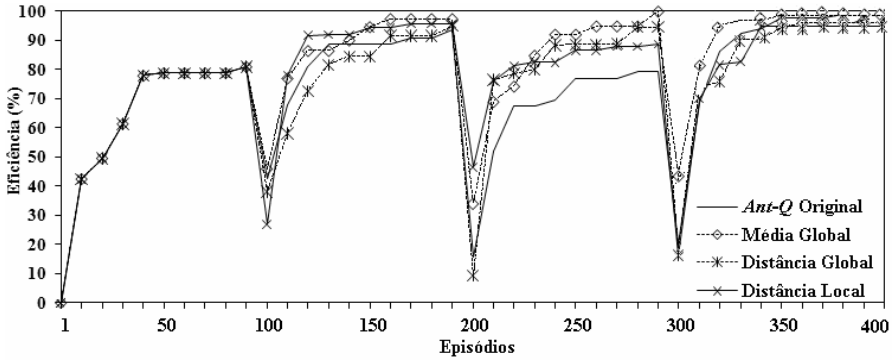


Figura 5.39: Campo limite de 1; 20% de alterações a cada 100 episódios

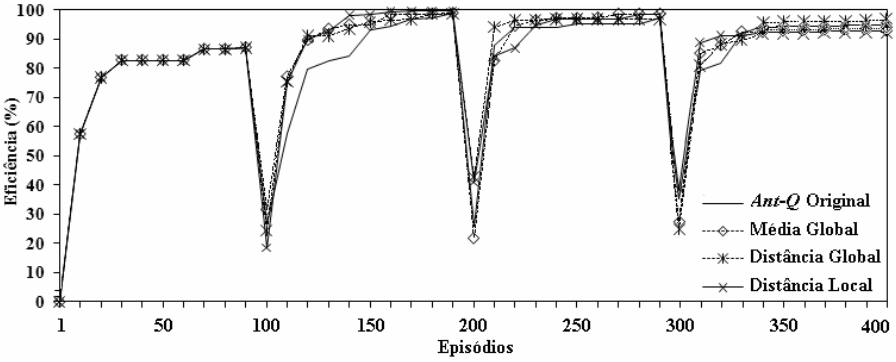


Figura 5.40: Campo limite de 2; 10% de alterações a cada 100 episódios

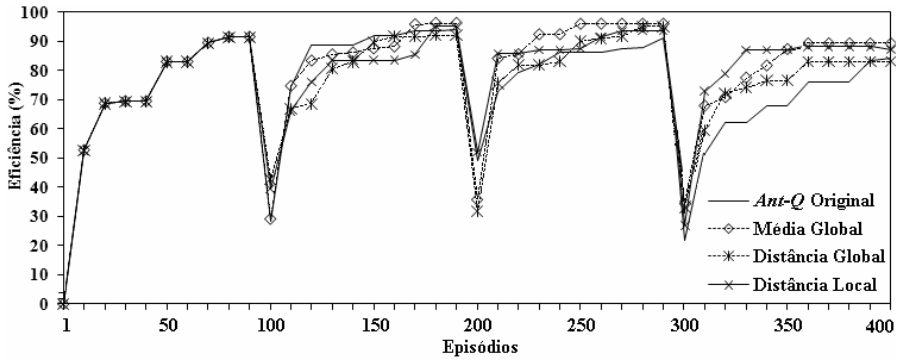


Figura 5.41: Campo limite de 2; 20% de alterações a cada 100 episódios

Uma observação interessante é o impacto do campo limite (adjacentes à posição cartesiana) nas estratégias. Mesmo com campo limite restrito, as estratégias melhoram a convergência do algoritmo. Em outros experimentos com o campo limite igual a 5, a eficiência do algoritmo *Ant-Q* é inferior (21%) quando comparado com a melhor estratégia (figuras 5.42 e 5.43).

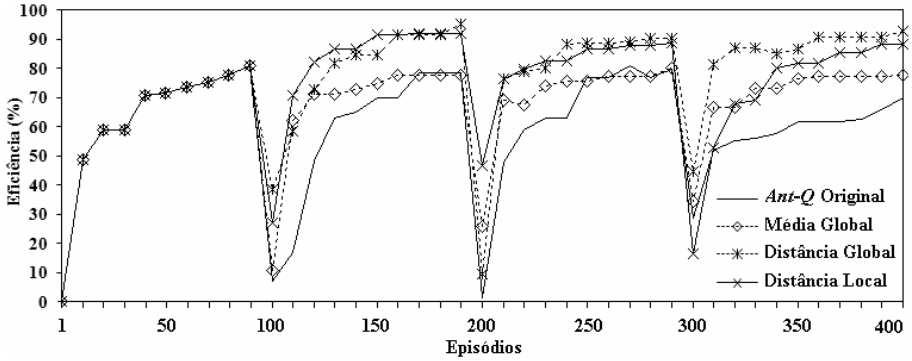


Figura 5.42: Campo limite de 5; 10% de alterações a cada 100 episódios

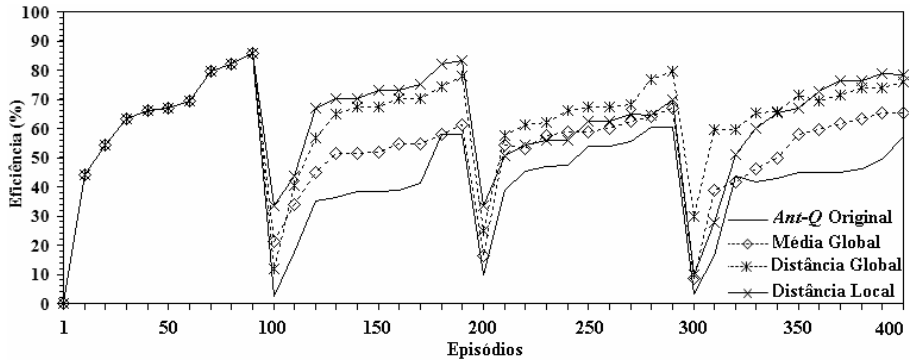


Figura 5.43: Campo limite de 5; 20% de alterações a cada 100 episódios

A estratégia média global é mais adequada quando o campo limite é inferior a 5 (figuras 5.38 a 5.41). Como a atualização é feita com a média de todos os valores de feromônio, o valor das ligações dos estados alterados é equalizado. As estratégias distância global e distância local podem convergir rapidamente quando o campo limite é igual 5 (figuras 5.42 e 5.43). Isso ocorre porque a atualização é proporcional à distância de cada ligação incidente nas ligações dos estados alterados. Assim, as ligações que não pertencem mais à política Q^*, terão o valor do feromônio enfraquecido.

Considerações finais

Esta seção apresentou um *framework* de teste para analisar o desempenho dos agentes com o algoritmo *Ant-Q* e para descrever o comportamento do *Ant-Q* com diferentes cenários, parâmetros e estratégias de atualização para ambientes dinâmicos. O *framework* apresentado é capaz de mostrar de maneira interativa o impacto da variação dos parâmetros e da quantidade de agentes no algoritmo, possibilitando conhecer os valores adequados dos parâmetros do *Ant-Q*.

Os resultados obtidos a partir da utilização das estratégias de atualização de políticas para ambientes dinâmicos mostram que o desempenho do algoritmo *Ant-Q* é superior ao desempenho da política global descoberta sem as estratégias. Apesar das particularidades de cada estratégia, os agentes conseguem melhorar a política com atualizações globais e locais, mostrando que as estratégias podem ser usadas em estruturas sociais dinâmicas, onde a estrutura de uma rede baseada em relações é alterada com a interação dos agentes, como apresentada na seção 5.3.

A aplicação das estratégias de atualização é uma oportunidade para as estruturas sociais dinâmicas, pois como as estratégias são baseadas em valores de recompensas de políticas passadas, a estrutura social construída com as relações dos indivíduos pode ser melhorada a partir das recompensas sociais, identificadas a partir do processo de interação dos indivíduos.

5.3 *SAnt-Q (Social Ant-Q)*: um algoritmo de otimização baseado em Colônia de Formigas, Aprendizagem por Reforço e Teorias Sociais

Nas seções anteriores deste capítulo, pôde-se observar que a sociabilidade é uma característica importante para a aprendizagem por reforço e que pode influenciar a coordenação dos agentes que interagem compartilhando recompensas, conforme foi observado na primeira eta-

pa da metodologia. Alguns modelos de coordenação entre agentes são baseados no princípio de reforço (algoritmo *Ant-Q*) e é portanto necessário estudar metodologias que se beneficiem dessa sociabilidade.

A sociabilidade desse algoritmo é baseada na descoberta colaborativa de valores de aprendizagem (feromônio) acumulados ao longo do processo de interação. Entretanto, neste trabalho acreditamos que estruturas e relacionamentos explícitos entre indivíduos podem ser identificados com a teoria das redes sociais, que contribui e fornece modelos fundamentados tanto teoricamente quanto matematicamente para a análise de situações que exigem a tomada de decisão coletiva. A sociabilidade, neste caso, é decorrente das interações sociais, que emergem de uma estrutura social mostrando o relacionamento entre os agentes e os comportamentos dos indivíduos que influenciam na tomada de decisão.

A coordenação para a tomada de decisão coletiva é melhorada quando um sistema social é formado com os melhores indivíduos (elitismo), onde aqueles com maior força influenciam os demais através de recompensas individuais ou coletivas, seguindo os princípios da aprendizagem por reforço.

Isso mostra a necessidade de estudar esses princípios com as redes sociais, para a formalização de um processo de construção de estruturas sociais de tomada de decisão com o objetivo de aprimorar a coordenação baseada em aprendizagem por reforço. Essa seção mostra como a utilização desses princípios melhora a coordenação dos agentes a partir de uma estrutura social extraída do conhecimento adquirido e das relações entre os agentes ao longo das interações.

Agentes que empregam algoritmos de formação de colônia de formigas interagem formando uma rede e compartilhando informações no intuito de alcançarem os objetivos individuais e coletivos propostos. Quando agentes interagem, o valor da recompensa e/ou feromônio depositado na ligação que conecta os estados é alterado, podendo modificar a política de ação corrente. Em ambientes onde a aprendizagem é dinâmica, o valor e a ligação entre os estados também são alterados continuamente pela política ou pelo modelo de coordenação, tornando o processo dinâmico.

Em ambientes com características sociais, os agentes podem ser influenciados por atitudes comportamentais de outros agentes. Tais atitudes estão relacionadas a modelos de sistemas sociais, que tentam descrever a estrutura das relações ou ligações com as métricas da análise das redes sociais.

Nesse contexto, algoritmos de colônia de formigas e redes sociais possuem características semelhantes, como por exemplo, a similaridade da estrutura formada por estados que

estão relacionados por alguma propriedade em comum. Algoritmos de colônia de formigas conectam os estados para resolver ou melhorar um problema de otimização, onde os agentes são conduzidos por heurísticas ou valores de retorno pela ação, ou ainda alguma influência externa do ambiente. Dessa maneira, os agentes interagem no ambiente a partir de influências dos demais agentes que compõem o sistema e procuram se relacionar ou seguir as ações dos agentes com melhor utilidade.

5.3.1 Redes Baseadas em Relações

Em um cenário na qual as relações entre os estados representam uma solução, o objetivo é encontrar o arranjo de relações que aumenta a utilidade da política. Isso consiste em um problema de otimização combinatória, que pode ser formalizado com ferramentas da teoria dos grafos. Seja G=(E,R) um grafo ponderado e simétrico, no qual E é o conjunto dos estados e R o conjunto das relações, os estados são conectados por ligações, que denotam alguma relação entre os estados conectados. Nesse caso, o custo de cada relação de G é um valor que indica a intensidade associada à relação. Agentes devem interagir para intensificar os valores das relações, formando a combinação que gera o melhor custo para a solução.

Segundo Radclife-Brown (1940) a interação entre os agentes de um sistema social descreve a estrutura que pode ser vista como uma rede de relações. Em nosso trabalho, a rede de relações (ou rede de relacionamento) é formada pelo conjunto de estados e as possíveis relações. Uma relação é definida como a ligação que conecta um par de estados da rede, onde as relações são fortalecidas ou enfraquecidas com técnicas da análise de redes sociais e algoritmos baseados em colônia de formigas.

A relação é modificada quando o agente realiza uma ação, movendo-se de um estado para outro. Quando tal ação ocorre, o valor da relação entre esses estados é alterado, através de valores de feromônio ou informações externas, como por exemplo, recompensas sociais geradas por alguma métrica social. As relações são, em geral, assimétricas e um estado pode estar relacionado com vários estados ao longo das interações.

A relação entre um par de estados constitui a díade, e o conjunto das díades irá constituir um grafo. A análise das redes sociais demonstra que a análise de uma díade só tem significado se considerado o conjunto das demais díades, pois uma rede de relacionamento é melhor interpretada quando incluídos todos os estados (Wasserman e Faust, 1994). A rede de relacionamento proposta pode assumir uma ou várias políticas de ação. Para o problema do caixeiro viajante, utilizado como estudo de caso, o custo de uma política é a soma das distâncias euclidianas entre os estados conectados, formando um conjunto de estados $E = \{e_1, e_2, ...,$

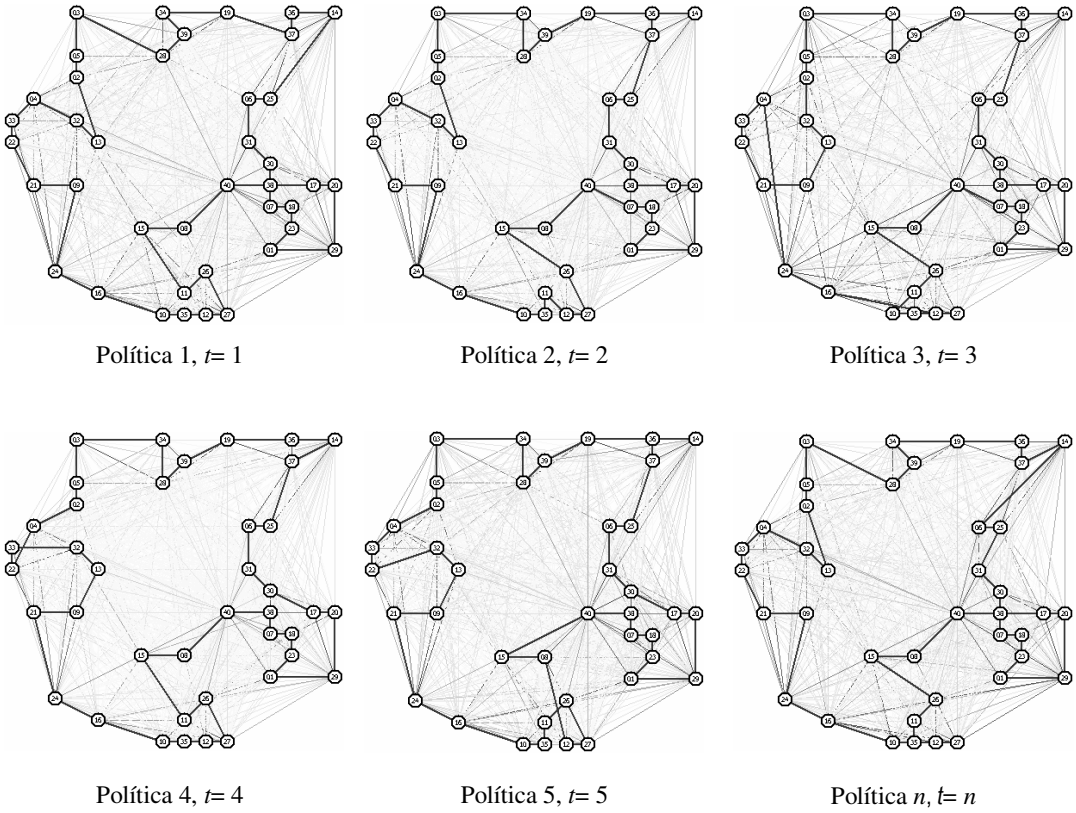
$e_n\}$ e um conjunto de relações $R = \{r_1, r_2, ..., r_n\}$, onde a quantidade possível de relações é |R|! e de políticas $\frac{|E|!}{2}$. O grau máximo de um estado para uma solução é 2 e o número máximo de relações dessa política é n-1, caracterizando um ciclo *hamiltoniano*. O custo da solução representa a soma das relações existentes na política encontrada.

Da rede de relacionamento proposta emerge uma topologia baseada na interação dos agentes, intensidade dos relacionamentos dos estados e em medidas de centralidade da análise de redes sociais. A análise das redes sociais descreve a topologia da rede, mostrando as relações, frequências e intensidade das relações durante a interação dos agentes no sistema. A estrutura da topologia forma a *rede de relacionamento*, e as interações dos agentes emergem como um sistema multiagente com características de redes sociais.

Uma rede de relacionamento vista como um sistema social é formada por agentes que interagem seguindo princípios da teoria do impacto social, como os conceitos da formação de opinião e da influência proposto por Latané (1981), e de outros modelos encontrados nas teorias sociais como a difusão da inovação (DeCanio e Watkins, 1998) e a formação de equipes (Abdallah e Lesser, 2004). Ao interagirem na rede, agentes estabelecem as relações e se tornam mais suscetíveis às influências de outros agentes quando adotam outras relações, mesmo quando todos possuem a mesma força de influência (*i.e.*, mesma probabilidade de se relacionarem). Nesse caso, as políticas de melhor utilidade têm maior probabilidade de influência, seguindo as regras das conexões preferenciais. Isso gera um processo de construção de rede de onde emergem demasiadamente estados com poucas ligações e poucos estados com muitas ligações (*hubs*), caracterizando uma rede livre de escala (Barabási *et al.* 2003).

De maneira intuitiva, agentes coordenados por algoritmos de colônia de formigas formam redes com comportamentos sociais, formadas por políticas de agentes que executam tarefas em conjunto. Esse modelo de rede pode ser analisado pela teoria do impacto social, estudada inicialmente por Latané (1981) que fornece elementos que explicam porque alguns agentes exercem maior influência em uma rede social do que os demais agentes.

Agentes interagem na rede na tentativa de descobrir uma política que satisfaça o problema. Quando uma solução é construída num episódio, é afirmado que uma política de ação foi descoberta. Normalmente em algoritmos de colônia de formigas, o agente com política de melhor utilidade dissemina com mais intensidade suas relações na rede. Na figura 5.44 são ilustradas algumas políticas de ação em uma rede de 40 estados, onde as linhas mais escuras são as relações mais intensas, indicando as políticas de melhor utilidade para a solução.



Política 1, t= 1 Política 2, t= 2 Política 3, t= 3

Política 4, t= 4 Política 5, t= 5 Política n, t= n

Figura 5.44: Exemplo de políticas de ação

Dado que um episódio pode produzir várias políticas, podem-se gerar muitas relações candidatas até a convergência. No entanto, apesar da quantidade de relações, não é assegurada a melhor política, devido a dependência dos agentes aos fatores de aprendizagem e coordenação. O efeito prático disso pode ser observado usando métricas da análise das redes sociais, onde se observou nos experimentos com o *Ant-Q* que a utilidade da política diminuía quando estados possuíam elevado grau de conectividade (muitas relações). O gráfico da figura 5.45 mostra que a eficiência da política aumenta quando o grau dos estados diminui, indicando que agentes egoístas não são suficientes, sendo necessário socializar. A constatação foi observada nos experimentos com um conjunto de problemas de *benchmark* (eil51 e eil71), apresentados nas seções seguintes.

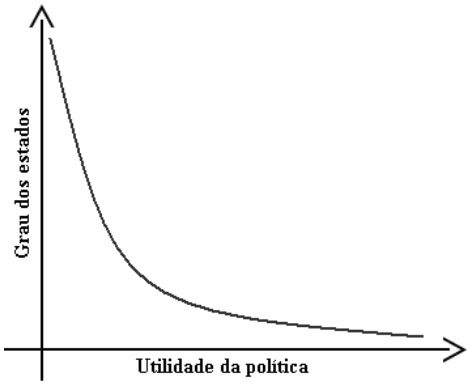


Figura 5.45: Eficiência da política em relação ao grau dos estados

Algoritmos de colônia de formigas são utilizados para coordenar o comportamento dos agentes de uma rede. É possível melhorar a coordenação dos agentes construindo uma rede de relacionamento com conceitos da teoria social e métricas bem conhecidas na análise das redes sociais, como intensidade das relações e medidas de centralidade. Utilizar algoritmos de otimização em redes com características sociais constitui uma nova abordagem que denominamos de *método de otimização social*, descrito nas seções seguintes.

5.3.2 Construção da Rede de Relacionamentos com o *SAnt-Q* (*Social Ant-Q*)

Baseado em medidas de centralidade e na intensidade da relação de estados díades, são apresentadas as etapas de construção da rede de relacionamentos com o *SAnt-Q*. É ilustrado na figura 5.46 o processo de crescimento da rede, usando os valores das políticas da rede de feromônio ($AQ(s,u)$) geradas pelo algoritmo *Ant-Q*.

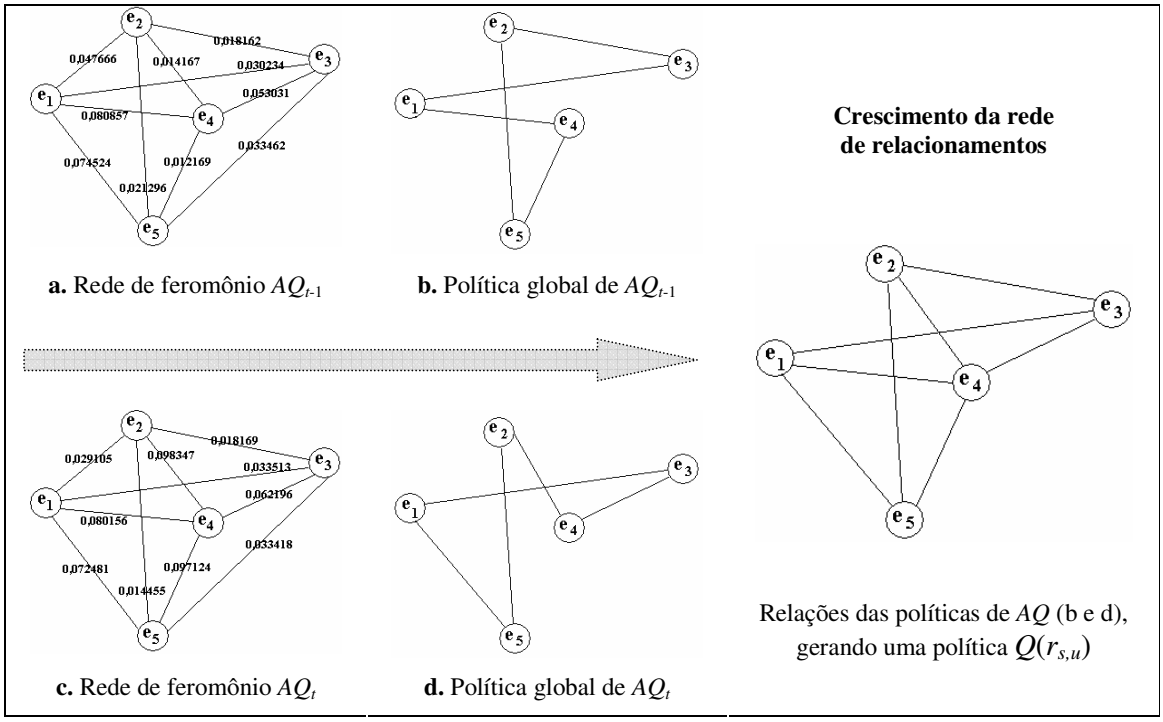


Figura 5.46: Processo de crescimento da rede de relacionamentos

A rede do *SAnt-Q* é construída a partir da política atual (AQ_t) e as políticas de episódios anteriores da rede de feromônio[5]. O processo de construção da rede com o *SAnt-Q* inicia quando os agentes completam o ciclo *hamiltoniano*, onde a quantidade de políticas geradas é igual ao número de agentes do sistema. Devido à grande quantidade de políticas geradas, são utilizadas apenas as políticas escolhidas para atualização global (aquelas que apresentam menor custo), diminuindo o número de políticas candidatas não satisfatórias. As imagens da figura 5.46 ilustram a política global atual (5.46d) e a política global anterior (5.46b), onde o custo dessas políticas é dado pela equação 5.8:

$$AQ(s,u)_t = \sum_{\forall r_{s,u} \in AQ} d(s,u) \tag{5.8}$$

no qual *r* é uma relação proveniente da política AQ_t do episódio *t* e *d*(s,u) é a distância euclidiana entre os estados conectados. O valor de AQ_t é utilizado para indicar a importância da política corrente. A equação 5.9 demonstra como os valores de AQ_{t-1} e AQ_t são usados para

[5] O *framework* para o desenvolvimento da rede de feromônio foi apresentado na seção 5.2.

calcular o valor que determina o fortalecimento ou enfraquecimento das relações da rede de relacionamento ($Q(r_{s,u})$).

$$\nu = \frac{AQ_{t-1}}{AQ_t} \tag{5.9}$$

O parâmetro *v* é utilizado como uma função de *fitness*, no qual é empregado na equação 5.10 para determinar a influência de *AQ* nas relações observadas na política atual.

$$\text{inf}(r) = (\nu \times r) - r \tag{5.10}$$

Computado o valor da função *inf*(*r*), a equação 5.11 determina o valor de $Q(r_{s,u})$, que armazena a intensidade da relação *r* que conecta os estados *s* e *u* dentro da rede de relacionamento:

$$Q(r_{s,u})_t = (Q(r_{s,u})_{t-1} + [\text{inf}(r_{s,u}) \times \rho]) \tag{5.11}$$

onde ρ é o parâmetro que indica a importância do valor da relação computada para a rede. É possível observar que quando $\nu \geq 1$, a solução candidata gerada é melhor do que a anterior, sendo que o caso contrário ocorre quando $\nu < 1$. Dessa forma, valores entre [0,1] foram simulados para o parâmetro ρ em diferentes situações: (i) quando $\nu \geq 1$ e a relação da díade está presente em AQ_t e AQ_{t-1} (a manutenção da díade possivelmente contribuiu para melhorar a solução); (ii) quando $\nu \geq 1$ e a relação da díade não está presente em AQ_{t-1} (o aparecimento da díade também contribuiu possivelmente para melhorar a solução); (iii) quando $\nu < 1$ e a relação da díade está presente em AQ_t e AQ_{t-1} (a manutenção da díade possivelmente piorou a solução); e (iv) quando $\nu < 1$ e a relação da díade não está presente em AQ_t (a solução atual piorou com a remoção da díade).

A relação de uma díade é dita presente quando essa pertence a AQ_t e AQ_{t-1} concomitantemente. A equação 5.12 mostra os valores usados para o parâmetro ρ. É considerado por conveniência que ρ= 0,8, ou seja, a influência da relação na atualização de Q(*r*) é de 80% quando a relação está presente em AQ_t e AQ_{t-1} e quando $\nu \geq 1$; ρ= 1 indica que a influência da relação é de 100%, quando não está presente em AQ_{t-1} e quando $\nu \geq 1$; ρ= 0,5 indica que a influência da relação é de 50%, quando está presente em AQ_t e AQ_{t-1} e quando $\nu < 1$. Final-

mente ρ= 1 quando a influência da relação é 100%, pois a relação da díade não está em AQ_t e $\nu < 1$. Essas configurações atribuem privilégios às relações que melhoram o custo da política global, fazendo emergir da rede uma estrutura capaz de melhorar as políticas, conforme discutido na subseção 5.4.3.

$$\rho = \begin{cases} 0{,}8 & se \quad (\nu \geq 1) \wedge (r_x \in AQ_t) \wedge (r_x \in AQ_{t-1}) \\ 1 & se \quad (\nu \geq 1) \wedge (r_x \in AQ_t) \wedge (r_x \notin AQ_{t-1}) \\ 0 & se \quad (\nu \geq 1) \wedge (r_x \notin AQ_t) \wedge (r_x \in AQ_{t-1}) \\ 0{,}5 & se \quad (\nu < 1) \wedge (r_x \in AQ_t) \wedge (r_x \in AQ_{t-1}) \\ 1 & se \quad (\nu < 1) \wedge (r_x \notin AQ_t) \wedge (r_x \in AQ_{t-1}) \\ 1 & se \quad (\nu < 1) \wedge (r_x \in AQ_t) \wedge (r_x \notin AQ_{t-1}) \\ 0 & se \quad ((\nu \geq 1) \vee (\nu < 1)) \wedge (r_x \notin AQ_t) \wedge (r_x \notin AQ_{t-1}) \end{cases} \tag{5.12}$$

O parâmetro ρ indica o quão rapidamente ocorre o fortalecimento ou enfraquecimento da relação. Se o valor de ρ é próximo de 0, o valor da $Q(r_{s,u})$ aumenta lentamente e com baixa convergência. Se o valor é 1, o algoritmo pode não convergir, pois o valor da relação é rapidamente fortalecido, induzindo os agentes ao mínimo global. O parâmetro ρ é similar ao fator de aprendizagem do *Q-learning*, no entanto, o valor de ρ é dinâmico, pois é usado conforme a melhora da solução, indicada pelo valor do parâmetro *v*. Os valores utilizados para ρ foram sugeridos a partir de constatações observadas nos experimentos, onde os valores foram testados e variados no intervalo de [0,1] em um conjunto de problemas de *benchmark* (eil51 e eil76).

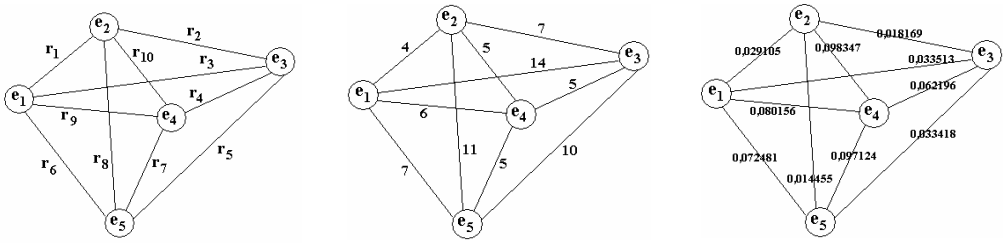
A. Demonstração Analítica

Nesta seção é apresentado um exemplo analítico de cálculo da metodologia apresentada anteriormente. Uma rede de estados conectados por valores de feromônio apresenta os seguintes componentes: um conjunto de estados E= $\{e_1, e_2, e_3, e_4, e_5\}$ e um conjunto de relações R= $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}\}$, onde a distância euclidiana entre os estados é: $d_{e1,e2}$=4, $d_{e1,e3}$=14, $d_{e1,e4}$=6, $d_{e1,e5}$=7, $d_{e2,e3}$=7, $d_{e2,e4}$=5, $d_{e2,e5}$=11, $d_{e3,e4}$=5, $d_{e3,e5}$=10, $d_{e4,e5}$= 5. Os valores de feromônio são substituídos pelas distâncias para indicar o custo da política global (melhor caminho). A tabela 5.3 mostra as relações, distâncias e feromônios dos estados conectados.

Tabela 5.3: Relações, distâncias e feromônios

Estados Conectados	Relação	Distância	Feromônio
e_1 e_2	r_1	4	0,029105
e_1 e_3	r_3	14	0,033513
e_1 e_4	r_9	6	0,080156
e_1 e_5	r_6	7	0,072481
e_2 e_3	r_2	7	0,018169
e_2 e_4	r_{10}	5	0,098347
e_2 e_5	r_8	11	0,014455
e_3 e_4	r_4	5	0,062196
e_3 e_5	r_5	10	0,033418
e_4 e_5	r_7	5	0,097124

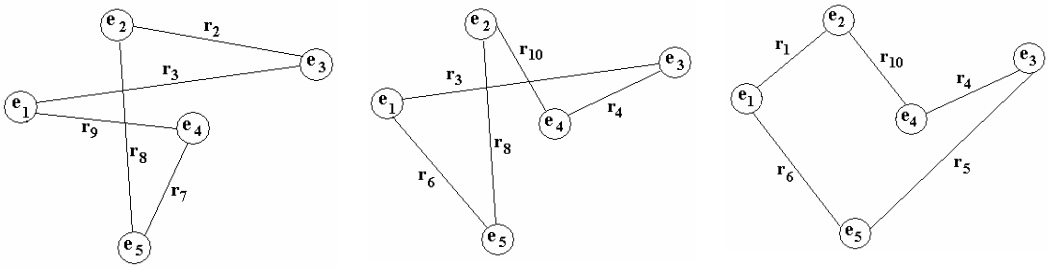
A figura 5.47 ilustra os valores da tabela 5.3 nos grafos.



a. Relações b. Distâncias c. Feromônios

Figura 5.47: Grafos de relações, distâncias e feromônios

Dados os subconjuntos R_1= {r_2, r_3, r_7, r_8, r_9} ∈ AQ_{t_1}, R_2= {r_3, r_4, r_6, r_8, r_{10}} ∈ AQ_{t_2} e R_3= {r_1, r_{10}, r_4, r_5, r_6} ∈ AQ_{t_3} ilustrados na figura 5.48, a rede de relacionamento é construída com as equações 5.8, 5.9, 5.10, 5.11 e 5.12 na qual as tabelas 5.4, 5.5 e 5.6 exemplificam os procedimentos. A rede é inicializada com os relacionamentos observados na política R_1 sendo os pesos inicializados arbitrariamente com valor 0,1.



AQ_{t_1} AQ_{t_2} AQ_{t_3}

Figura 5.48: Políticas usadas para simular o crescimento da rede de relacionamento

Usando a equação 5.8, o valor do ciclo *hamiltoniano* das políticas AQ_{t_1}, AQ_{t_2} e AQ_{t_3} da figura 5.48 são computados, onde: AQ_{t_1}= 43; AQ_{t_2}= 42; AQ_{t_3}= 31. Com esses valores, são calculados com a equação 5.9 os valores para *v*, donde: $v_1 = \frac{AQ_{t_1}}{AQ_{t_2}} = \frac{43}{42} = 1{,}023$ e $v_2 = \frac{AQ_{t_2}}{AQ_{t_3}} = \frac{42}{31} = 1{,}354$.

Com os valores de v_1 e v_2 é possível computar a influência nas relações com a equação 5.10. As tabelas 5.4 e 5.6 mostram a influência de v_1 e v_2 nas relações. A rede de relacionamentos possui uma tabela de aprendizagem $Q{:}(R) \rightarrow \Re$, que indica as relações e os valores de intensidade de cada relação *r* de *R*. Para estimar *Q*(*r*) é utilizada a equação 5.11 que determina a intensidade das relações na rede. As tabelas 5.5 e 5.7 mostram a equação 5.11 computando a intensidade das relações de *Q*(*r*) em t_2 e t_3.

Tabela 5.4: Influência de v_1 nas relações de *Q*(*r*) em t_2

	Influência de v_1= 1,023	
R{ }	*Q*(*r*) em t_1 (valores arbitrários)	Equação 5.10: $inf(r_x) = (v_1 \times r_x) - r_x$
r_1	$Q(r_1)$= 0,1	$inf(r_1)$= (1,023 x 0,1) – 0,1= 0,0023
r_2	$Q(r_2)$= 0,1	$inf(r_2)$= (1,023 x 0,1) – 0,1= 0,0023
r_3	$Q(r_3)$= 0,1	$inf(r_3)$= (1,023 x 0,1) – 0,1= 0,0023
r_4	$Q(r_4)$= 0,1	$inf(r_4)$= (1,023 x 0,1) – 0,1= 0,0023
r_5	$Q(r_5)$= 0,1	$inf(r_5)$= (1,023 x 0,1) – 0,1= 0,0023
r_6	$Q(r_6)$= 0,1	$inf(r_6)$= (1,023 x 0,1) – 0,1= 0,0023
r_7	$Q(r_7)$= 0,1	$inf(r_7)$= (1,023 x 0,1) – 0,1= 0,0023
r_8	$Q(r_8)$= 0,1	$inf(r_8)$= (1,023 x 0,1) – 0,1= 0,0023
r_9	$Q(r_9)$= 0,1	$inf(r_9)$= (1,023 x 0,1) – 0,1= 0,0023
r_{10}	$Q(r_{10})$= 0,1	$inf(r_{10})$= (1,023 x 0,1) – 0,1= 0,0023

A tabela 5.5 mostra que se a relação de AQ_{t_1} não está presente na AQ_{t_2}, o valor da relação não é alterado (r_1, r_2, r_5, r_7, r_9), pois é atribuído a ρ o valor 0.

Tabela 5.5: Intensidade das $Q(r)$ em t_2

R{ }	Computa $Q(r)$		
	$Q(r)$ em t_1	Equação 5.11: $Q(r_{s,u})_t = (Q(r_{s,u})_{t-1} + [\inf(r_{s,u}) \times \rho])$	ρ $(\nu \geq 1)$
r_1	$Q(r_1)$= 0,1	$Q(r_1)$= 0,1 + (0,0023 x 0)= **0,1**	0
r_2	$Q(r_2)$= 0,1	$Q(r_2)$= 0,1 + (0,0023 x 0)= **0,1**	0
r_3	$Q(r_3)$= 0,1	$Q(r_3)$= 0,1 + (0,0023 x 0,8)= **0,1018**	0,8
r_4	$Q(r_4)$= 0,1	$Q(r_4)$= 0,1 + (0,0023 x 1)= **0,1023**	1
r_5	$Q(r_5)$= 0,1	$Q(r_5)$= 0,1 + (0,0023 x 0)= **0,1**	0
r_6	$Q(r_6)$= 0,1	$Q(r_6)$= 0,1 + (0,0023 x 1)= **0,1023**	1
r_7	$Q(r_7)$= 0,1	$Q(r_7)$= 0,1 + (0,0023 x 0)= **0,1**	0
r_8	$Q(r_8)$= 0,1	$Q(r_8)$= 0,1 + (0,0023 x 0,8)= **0,1018**	0,8
r_9	$Q(r_9)$= 0,1	Q(r_9)= 0,1 + (0,0023 x 0)= **0,1**	0
r_{10}	Q(r_{10})= 0,1	Q(r_{10})= 0,1 + (0,0023 x 1)= **0,1023**	1

O procedimento seguinte é calcular a influência de v_2 nas relações de AQ_{t_3} (tabela 5.6).

Tabela 5.6: Influência de v_2 nas relações de $Q(r)$ em t_3

R{ }	Influência de v_2= 1,354	
	$Q(r)$ em t_2	Equação 5.10 $inf(r_x) = (v_1 \mathbf{x} r_x) - r_x$
r_1	$Q(r_1)$= 0,1	$inf(r_1)$= (1,354 x 0,1) – 0,1= **0,0354**
r_2	$Q(r_2)$= 0,1	$inf(r_2)$= (1,354 x 0,1) – 0,1= **0,0354**
r_3	$Q(r_3)$= 0,1018	$inf(r_3)$= (1,354 x 0,1018) – 0,1018= **0,0360**
r_4	$Q(r_4)$= 0,1023	$inf(r_4)$= (1,354 x 0,1023) – 0,1023= **0,0362**
r_5	$Q(r_5)$= 0,1	$inf(r_5)$= (1,354 x 0,1) – 0,1= **0,0354**
r_6	$Q(r_6)$= 0,1023	$inf(r_6)$= (1,354 x 0,1023) – 0,1023= **0,0362**
r_7	$Q(r_7)$= 0,1	$inf(r_7)$= (1,354 x 0,1) – 0,1= **0, 0354**
r_8	$Q(r_8)$= 0,1018	$inf(r_8)$= (1,354 x 0,1018) – 0,1018= **0,0360**
r_9	$Q(r_9)$= 0,1	$inf(r_9)$= (1,354 x 0,1) – 0,1= **0, 0354**
r_{10}	$Q(r_{10})$= 0,1023	$inf(r_{10})$= (1,354 x 0, 1023) – 0,1023= **0,0362**

É possível observar na tabela 5.7 que os valores das relações r_2, r_3, r_7, r_8 e r_9 não são alterados, pois essas relações não estão presentes em AQ_{t_3}. Na tabela 5.8 são mostrados os valores da rede de relacionamentos gerada a partir dos episódios t_1, t_2 e t_3.

Tabela 5.7: Intensidade das $Q(r)$ em t_3

R{ }	Computa $Q(r)$		
	$Q(r)$ em t_2	Equação 5.11: $Q(r_{s,u}) = (Q_{t-1}(r_{s,u}) + [\inf(r_{s,u}) \times \rho])$	ρ ($\nu \geq 1$)
r_1	$Q(r_1)$= 0,1	$Q(r_1)$= 0,1 + (0,0354 x 1)= **0,1354**	1
r_2	$Q(r_2)$= 0,1	$Q(r_2)$= 0,1 + (0,0354 x 0)= **0,1**	0
r_3	$Q(r_3)$= 0,1018	$Q(r_3)$= 0,1018 + (0,0360 x 0)= **0,1018**	0
r_4	$Q(r_4)$= 0,1023	$Q(r_4)$= 0,1023 + (0,0362 x 0,8)= **0,1312**	0,8
r_5	$Q(r_5)$= 0,1	$Q(r_5)$= 0,1 + (0,0354 x 1)= **0,1354**	1
r_6	$Q(r_6)$= 0,1023	$Q(r_6)$= 0,1023 + (0,0362 x 0,8)= **0,1312**	0,8
r_7	$Q(r_7)$= 0,1	$Q(r_7)$= 0,1 + (0,0354 x 0)= **0,1**	0
r_8	$Q(r_8)$= 0,1018	$Q(r_8)$= 0,1018 + (0,0360 x 0)= **0,1018**	0
r_9	$Q(r_9)$= 0,1	$Q(r_9)$= 0,1 + (0,00354 x 0)= **0,1**	0
r_{10}	$Q(r_{10})$= 0,1023	Q(r_{10})= 0,1023 + (0,0362 x 0,8)= **0,1312**	0,8

Na figura 5.49 é ilustrada a rede de relacionamento com os valores das relações dos 3 episódios. As linhas na cor mais escura indicam as relações mais intensas (melhor política) da rede de relacionamentos no episódio t_3. Para selecionar a melhor polícia foi usado o algoritmo de busca *Best-first*, descrito em Pearl (1984).

Tabela 5.8: Valores da $Q(r)$ em t_1, t_2 e t_3

R{ }	Valores de intensidade das $Q(r)$		
	$Q(r)$ de AQ_{t_1}	$Q(r)$ de AQ_{t_2}	$Q(r)$ de AQ_{t_3}
r_1	0,1	0,1	0,1354
r_2	0,1	0,1	0,1
r_3	0,1	0,1018	0,1018
r_4	0,1	0,1023	0,1312
r_5	0,1	0,1	0,1354
r_6	0,1	0,1023	0,1312
r_7	0,1	0,1	0,1
r_8	0,1	0,1018	0,1018
r_9	0,1	0,1	0,1
r_{10}	0,1	0,1023	0,1312

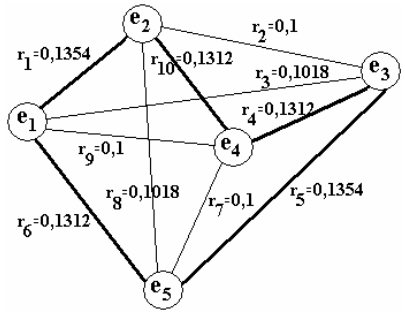


Figura 5.49: Rede de relacionamentos em t_3

O processo da construção da rede é iterativo e ocorre até que uma condição de parada do algoritmo seja satisfeita.

Um dos problemas observados com a rede de relacionamentos é que nos episódios iniciais os agentes não conseguem estabilizar suas relações quando utilizam os valores de $Q(r)$, causando variações no custo das políticas. Isso ocorre porque a intensidade dos valores das relações no início da aprendizagem ainda é baixa, na intenção de evitar a estagnação. Outra observação, é que algoritmos baseados em recompensas realizam a busca no espaço de estados usando regras de transição como estratégias de exploração, o que altera com frequência o relacionamento entre os estados no período inicial da aprendizagem.

Uma característica importante da rede de relacionamentos é o crescimento da rede sem a dependência e a quantidade de parâmetros de aprendizagem e atualizações globais. Isso é importante, pois é devidamente sabido que algoritmos como o *Ant-Q* e o *Q-learning* são sensíveis a estes parâmetros e aos dados do domínio, como por exemplo, o posicionamento dos estados no plano, e a quantidade de indivíduos na rede, conforme discutido na seção de análise do *Ant-Q* (seção 5.2). Outra observação importante é que como a rede de relacionamentos não depende do posicionamento dos estados (função heurística), a exploração é favorecida quando alguma relação é removida ou inserida.

Outra questão observada com a rede de relacionamentos é a possibilidade de ocorrer pouca frequência de interações entre alguns estados, devido a influência dos valores da rede de feromônio. Isso ocorre, porque o *Ant-Q* utiliza os valores de feromônios que foram estimados pelas atualizações locais e globais (equações 5.3 e 5.4), onde normalmente, esses valores são diminuídos (evaporados) para melhorar a exploração. Esse comportamento ocasiona a existência de algumas relações com valores muito baixos, o que pode não garantir a convergência do algoritmo em alguns momentos da interação.

Uma alternativa foi adaptarmos o *Ant-Q* para que ele utilize após um determinado número de episódios a rede de feromônio e a rede de relacionamentos, denominando esta nova

abordagem de *SAnt-Q* (*Social Ant-Q*). Para mostrar o processo de interação do algoritmo de colônia de formigas com os princípios da análise das redes sociais, é ilustrado na figura 5.50 o diagrama de atividades.

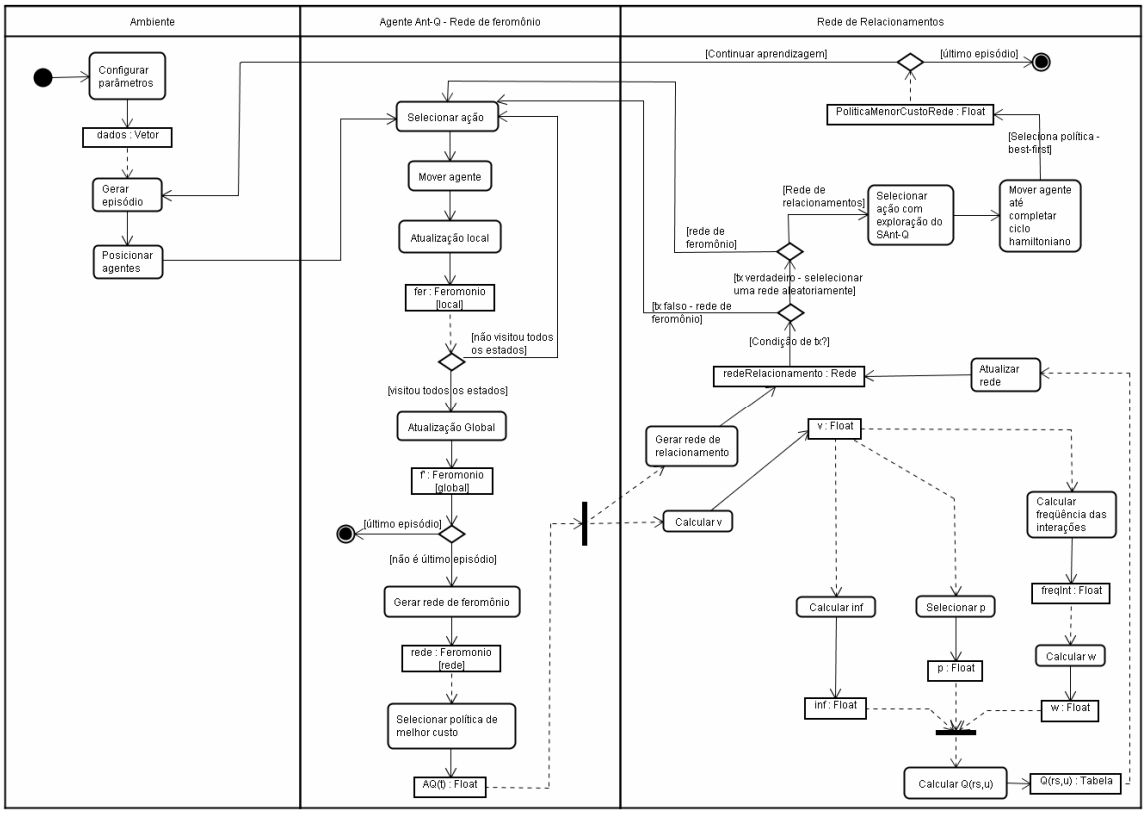


Figura 5.50: Diagrama de atividades

Na primeira coluna (ambiente) são configurados os parâmetros de aprendizagem do algoritmo *Ant-Q* e os valores de configuração do ambiente. Na sequência, um episódio é gerado e os agentes são posicionados aleatoriamente nos estados. Na segunda coluna (agente *Ant-Q* - rede de feromônio) são gerados os valores de recompensas pelo algoritmo *Ant-Q* para a construção da rede de feromônio. Quando essa rede é gerada, é possível selecionar com a equação 5.8 a política de melhor custo investigando as políticas candidatas do episódio corrente. Na terceira coluna (rede de relacionamentos) são mostradas as atividades que dão início à construção da rede de relacionamentos, na qual os procedimentos de tais atividades foram descritos nesta subseção. A rede de relacionamentos é utilizada pelo *Ant-Q* quando um episódio t_x é alcançado. A partir desse episódio, a probabilidade do agente explorar a rede de relacionamentos ou a rede de feromônio é igual. Isso melhora a exploração, pois o agente utiliza tanto a estratégia de exploração do *Ant-Q* como a estratégia do *SAnt-Q* (equação 5.15).

Para melhorar a exploração do *SAnt-Q* na rede de relacionamentos, a regra de transição do *Ant-Q* foi adaptada, já que o *SAnt-Q* não disponibiliza uma função heurística baseada nas distâncias associadas às relações ($r_{s,u}$). Com a regra de transição do *SAnt-Q* o agente posicionado em um estado *s* move-se para o estado *u* usando a regra da equação 5.15:

$$u = \begin{cases} \arg\max(Q(r_{s,u})_t) & se \quad q_0 = 1 \\ I & se \quad q_0 = 0 \end{cases} \tag{5.15}$$

onde argmax($Q(r_{s,u})_t$) indica a relação de maior intensidade no estado *s* no episódio *t* e *I* é a probabilidade de selecionar a relação *u* de acordo com a equação 5.16:

$$I = \frac{Q(r_{s,u})_t}{\sum_{i=1 | i \in Adjacentes(s)}^{x} Q(r_{s,i})_t} \tag{5.16}$$

Com q_0= 1 a escolha da relação *u* é similar ao *exploration*, onde a probabilidade do agente selecionar um novo estado para se mover é proporcional ao valor das relações com os estados adjacentes. Quando q_0= 0 é selecionada a ação gerada pelo maior relacionamento. O parâmetro q_0 é selecionado arbitrariamente.

Na subseção a seguir, apresentamos os resultados experimentais comparando o *Ant-Q* e o *SAnt-Q*.

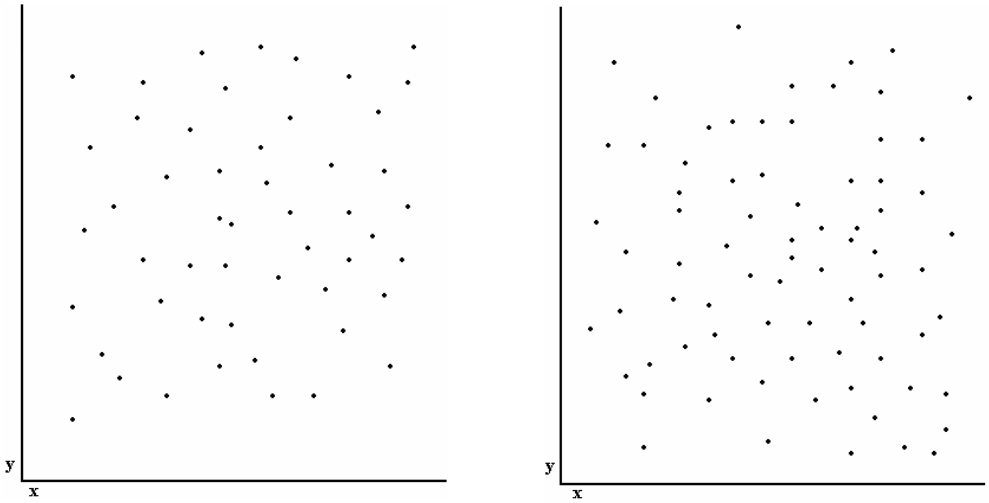
5.3.3 Resultados Experimentais

Nos experimentos foi observada a utilidade das políticas geradas com agentes interagindo na rede de feromônio do *Ant-Q* e na rede de relacionamentos do *SAnt-Q*. Os experimentos foram realizados em problemas de *benchmark*: eil51 e eil76, encontrados na biblioteca *online* TSPLIB[6] (Reinelt, 1991).

Os conjuntos eil51 e eil76 são compostos por 51 e 76 estados respectivamente e foram formulados por Christofides e Eilon (1969). Tais conjuntos representam características importantes para simular problemas de otimização combinatorial, como por exemplo, a quantidade de estados e a existência de estados adjacentes com distâncias semelhantes. Eles também fo-

[6] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

ram utilizados em (Dorigo, 1992; Gambardella e Dorigo, 1995; Bianchi *et al.* 2002). A figura 5.51 mostra a distribuição dos estados no plano, onde estão expressos em um sistema euclidiano de coordenadas 2D.



a. Conjunto de instâncias (eil51) b. Conjunto de instâncias (eil76)

Figura 5.51: Distribuição dos estados no plano

Os experimentos foram rodados 10 vezes para cada conjunto. Os parâmetros do algoritmo *Ant-Q* foram configurados com os seguintes valores: δ= 1; β= 2; γ= 0,3; α= 0,1; q_0= 0,9 e W= 10. A quantidade de agentes é igual ao número de estados de cada conjunto. Para observar o impacto da rede de relacionamentos na política com o *SAnt-Q*, a rede foi utilizada a partir dos episódios t_{30}, t_{50} e t_{100}.

Foram utilizadas como critério de parada as quantidades de 500, 5000 e 10000 episódios. Vale observar que devido ao número de estados e a complexidade dos problemas, a quantidade de episódios não é suficiente para encontrar a melhor política. No entanto, o objetivo dos experimentos é avaliar o impacto da rede de relacionamentos no algoritmo *Ant-Q* e na utilidade da política final com o *SAnt-Q*.

Notamos que para avaliar o desempenho de uma técnica pode-se empregar diferentes métricas, como o tempo de execução, a quantidade de episódios da melhor política ou considerar somente a utilidade das melhores políticas encontradas.

Para limitar a quantidade dos experimentos, foi considerada a utilidade das políticas em um número estabelecido de episódios, observando então a política com melhor custo no final da aprendizagem. Ademais, vale observar que há diferença entre encontrar a política ótima e encontrar uma política satisfatória. Encontrar a política ótima, normalmente significa

empregar buscas exaustivas, pois é a maneira mais comum de explorar todo o espaço de estados. Por outro lado, encontrar uma política satisfatória, significa encontrar uma alternativa que satisfaça o problema, sem se importar se a melhor política descoberta é a melhor possível.

Para verificar se houve ou não diferença significativa dos algoritmos, foi escolhido um teste estatístico do tipo não-paramétrico, devido às características dos experimentos e por ser mais provável de rejeitar a hipótese nula (Siegel, 1975). Assim, os testes estatísticos não-paramétricos (*e.g.*, teste de Friedman) não têm exigências quanto ao conhecimento da distribuição da variável na população, onde são testadas associações, dependência/independência e modelos ao invés de parâmetros. Para o teste de Friedman que escolhemos, os algoritmos são ranqueados para cada conjunto de dados separadamente, onde o algoritmo com melhor desempenho ocupa a primeira posição do ranque, o segundo melhor ocupa a segunda posição no ranque e assim sucessivamente (Demsar, 2006). Em caso de empates no desempenho dos algoritmos *Ant-Q* e *SAnt-Q*, é feita a média dos ranques.

Deste modo, o teste de Friedman com as equações 5.13 e 5.14 computa o ranque e a média do custo das políticas, onde r_i^j é o ranque do *j*-ésimo dentre dos *k* algoritmos dos *N* conjuntos de dados.

$$R_j = \frac{1}{N}\sum_i r_i^j \tag{5.13}$$

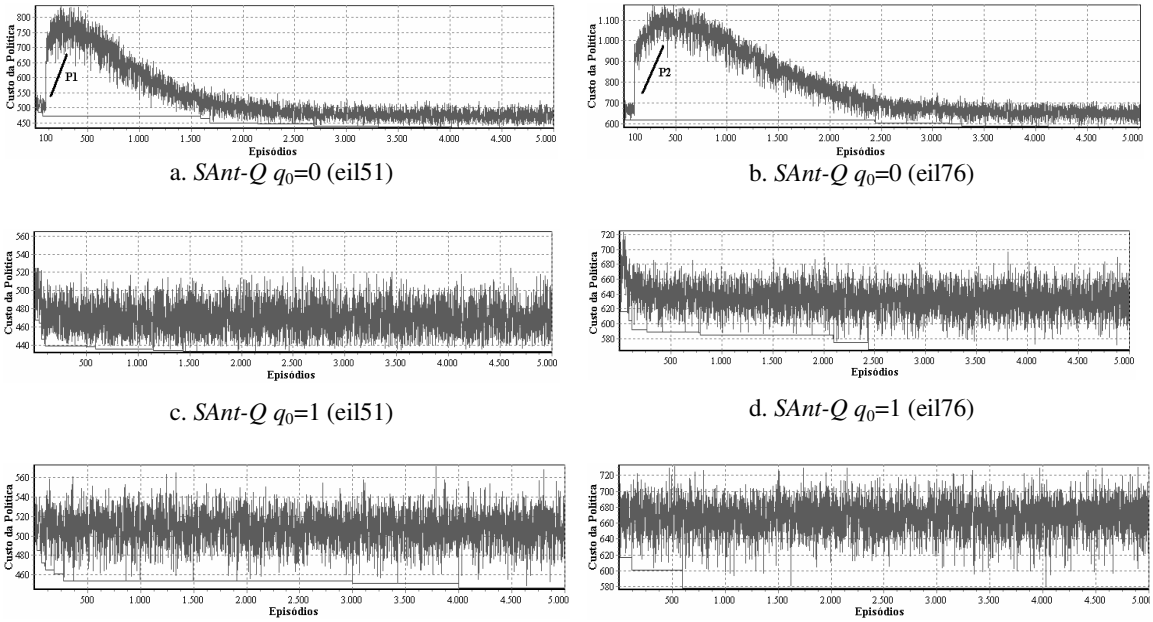
$$\chi_F^2 = \frac{12}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)}{4}\right] \tag{5.14}$$

O objetivo deste teste é verificar se os algoritmos apresentam diferenças significativas. Caso a hipótese nula seja caracterizada, o custo das políticas dos algoritmos é equivalente uma vez que eles possuem ranques iguais. Ao considerar como hipótese nula a inexistência de diferenças entre as condições dos *k* algoritmos, se obtêm amostras bem distribuídas, não havendo co-relação entre elas. Porém, para verificar se há correlação entre as condições, deve-se fazer o somatório das variâncias (Q) dos ranques. Obtido o valor de Q, calcula-se o *p-valor* como a probabilidade do valor ser superior ou igual à variância obtida utilizando a distribuição qui-quadrada com *k*-1 graus de liberdade.

O resultado numérico do teste estatístico de Friedman fornece um nível de significância (*p-valor*). Caso este seja menor que 0,05 (valor usado nos experimentos) 5% será a tole-

rância de aceitação, então é recomendável rejeitar a hipótese nula, podendo afirmar que existe diferença significativa entre os experimentos.

Os resultados iniciais apresentam o impacto da regra de transição do *SAnt-Q*. Nos experimentos utilizando a regra de transição com o parâmetro q_0= 0 foram necessários muitos episódios (aproximadamente 5000) para encontrar as melhores políticas nos problemas eil51 e eil76, conforme os gráficos da figura 5.52. Os picos P1 e P2 observados no episódio 100 das figuras 5.52a e 5.52b, são decorrentes dos valores da rede de feromônio apresentarem relações que ainda foram pouco exploradas pelo *Ant-Q*. Em ambientes com muitos estados, e.g. eil51 e eil76, o agente necessita de um número maior de episódios, devido à necessidade de interagir com cada estado inúmeras vezes para realizar o aprendizado.



a. *SAnt-Q* q_0=0 (eil51)

b. *SAnt-Q* q_0=0 (eil76)

c. *SAnt-Q* q_0=1 (eil51)

d. *SAnt-Q* q_0=1 (eil76)

e. *Ant-Q* (eil51)

f. *Ant-Q* (eil76)

Figura 5.52: Variações do custo das políticas com o *Ant-Q* e o *SAnt-Q*

Apesar das variações do custo das políticas do *SAnt-Q* com q_0= 1 (figuras 5.52c e 5.52d), as variações são menores quando comparadas com as políticas do *Ant-Q* (figuras 5.52e e 5.52f). Devido a essas observações, os resultados com *SAnt-Q* foram obtidos com o parâmetro q_0= 1. No entanto, vale observar que o custo médio das melhores políticas (tabelas 5.9 e 5.10) com q_0= 0 é menor que o apresentado com q_0= 1 com 5.000 episódios. No problema eli51, o custo médio das melhores políticas com q_0= 0 é em média, 0,98% menor com t_{30}; 0,99% com t_{50}; e 0,98% com t_{100} (tabela 5.9) do que o custo daquelas geradas com q_0= 1. No

problema eil76, o custo médio das melhores políticas com q_0= 0 é em média, 0,98% menor com t_{30}; 0,97% com t_{50}; e 0,96% com t_{100} (tabela 5.10) do que as produzidas com q_0= 1.

Tabela 5.9: Custo médio das melhores políticas (eil51) com 5000 episódios

	Ant-Q	*SAnt-Q*					
		q_0= 1			q_0= 0		
		t_{30}	t_{50}	t_{100}	t_{30}	t_{50}	t_{100}
Custo médio das políticas	455,67	436,73	438,71	441,46	432,1	434,9	436,4

Tabela 5.10: Custo médio das melhores políticas (eil76) com 5000 episódios

	Ant-Q	*SAnt-Q*					
		q_0= 1			q_0= 0		
		t_{30}	t_{50}	t_{100}	t_{30}	t_{50}	t_{100}
Custo médio das políticas	602,92	583,49	579,43	579,87	575,1	567,4	561,2

Uma explicação para os resultados preliminares observados é que após longo período de aprendizagem, as relações mais fortes são enfraquecidas, o que melhora a probabilidade de selecionar novos estados desejáveis. Isso é decorrente da configuração do parâmetro ρ, que favorece o crescimento da rede de relacionamentos com uma topologia de baixa densidade, o que diminui a instabilidade dos estados e a existência de *hubs*.

As tabelas 5.11 e 5.12 apresentam o custo das políticas com o *SAnt-Q* e o *Ant-Q*. É possível observar que o custo médio das políticas é melhor com o *SAnt-Q* quando comparado com o *Ant-Q*.

Tabela 5.11: Custo das políticas (eil51)

Experimentos	*Ant-Q*	*SAnt-Q* q_0=1; 500 episódios		
		t_{30}	t_{50}	t_{100}
1	453,93	431,61	441,25	451,98
2	453,70	438,99	439,89	438,57
3	455,76	443,53	437,79	437,69
4	456,09	440,51	439,48	445,30
5	454,36	431,72	431,42	433,43
6	451,65	436,30	436,64	445,92
7	455,24	434,62	441,93	431,37
8	455,39	437,31	445,28	440,06
9	460,94	440,86	438,26	445,23
10	459,67	431,90	435,20	445,13
Média	**455,67** (±2,768)	**436,73** (±4,246)	**438,71** (±3,840)	**441,46** (±6,357)

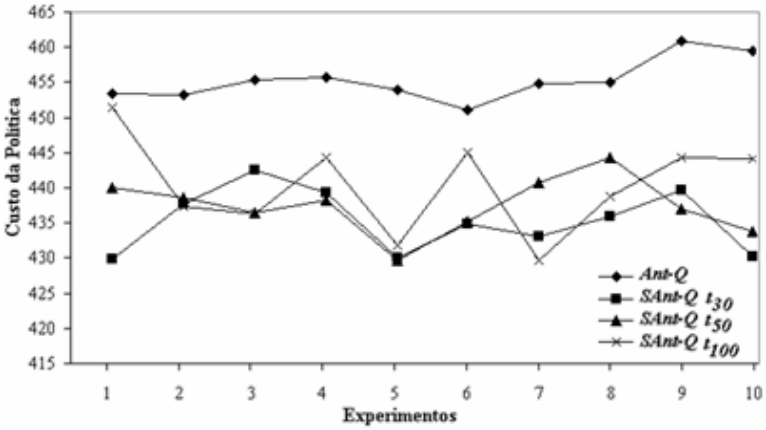


Figura 5.53: *Ant-Q* vs. *SAnt-Q*, eil51 com 500 episódios

Tabela 5.12: Custo da política (eil76)

Experimentos	*Ant-Q*	*SAnt-Q* q_0=1; 500 episódios		
		t_{30}	t_{50}	t_{100}
1	601,13	589,16	588,95	582,39
2	605,22	582,85	578,87	578,15
3	602,66	575,67	578,52	582,97
4	606,89	575,61	579,19	578,91
5	592,60	575,97	588,83	570,58
6	610,17	584,48	565,13	583,26
7	602,45	585,33	581,66	579,24
8	599,15	583,56	584,36	578,53
9	603,03	583,95	576,69	584,82
10	605,98	598,38	572,18	579,93
Média	**602,92** (±4,798)	**583,49** (±6,981)	**579,43** (±7,257)	**579,87** (±4,004)

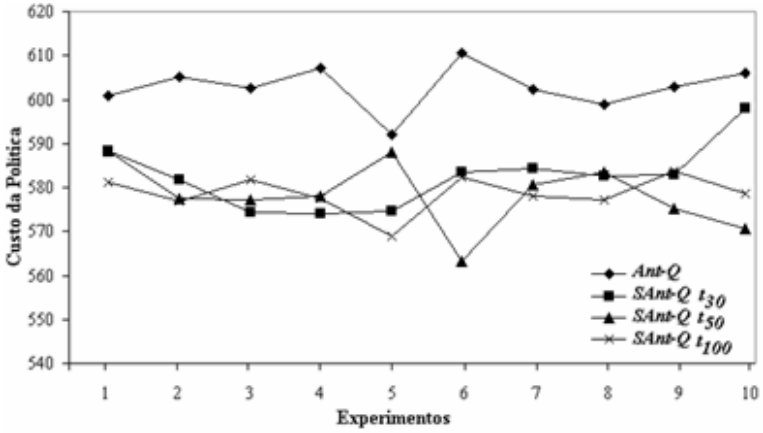


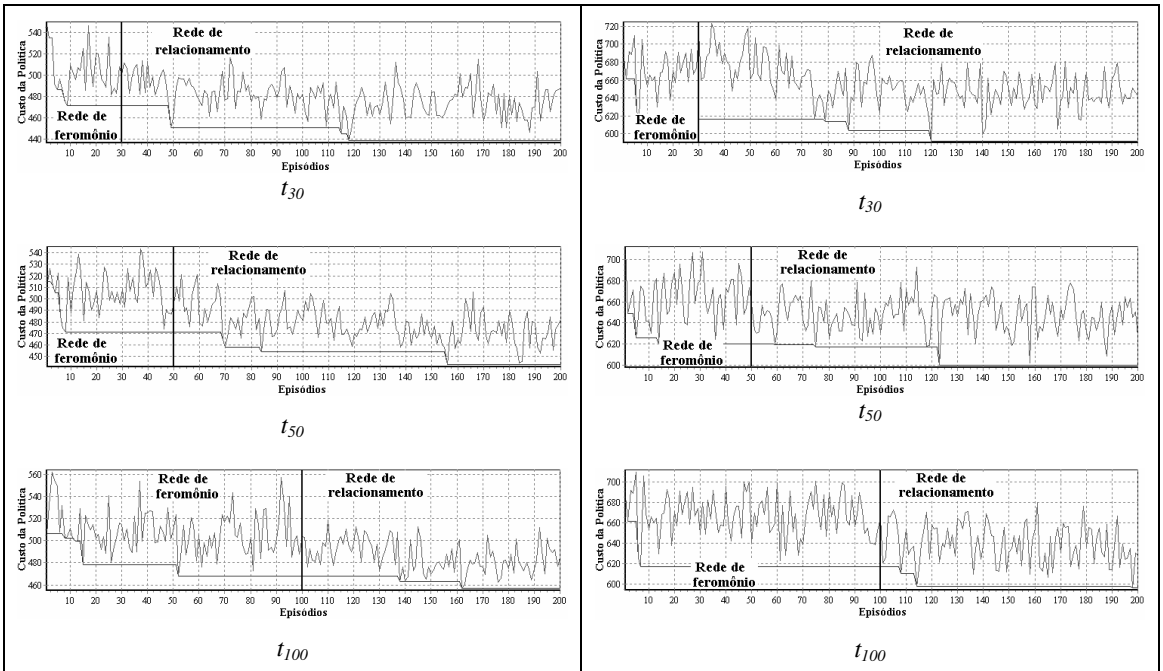
Figura 5.54: *Ant-Q* vs. *SAnt-Q*, eil76 com 500 episódios

O custo médio das melhores políticas com o *SAnt-Q* no problema eli51 é, em média, 4,14% menor com t_{30}; 3,72% com t_{50}; e 3,11% com t_{100} quando comparado com o custo médio da política do *Ant-Q* (tabela 5.11). No problema eil76 o custo médio das políticas do *SAnt-Q* é, em média, 3,22% menor com t_{30}; 3,89% com t_{50}; e 3,82% com t_{100} (tabela 5.12) quando comparadas com as políticas do *Ant-Q*. A melhora é decorrente do processo de evolução da rede de relacionamentos que influencia a exploração do algoritmo, tendo impacto positivo no algoritmo de otimização.

Pode-se observar que ao utilizar a rede de relacionamentos após os episódios t_{50} e t_{100}, o algoritmo *SAnt-Q* precisa de mais episódios para melhorar o custo da política. Isso ocorre porque a rede de relacionamentos está formada com uma quantidade maior de políticas, aumentando assim o grau dos estados, e consequentemente, o espaço de busca e a possibilidade

de novas relações. Em outras palavras, a rede tem maior densidade, o que necessariamente não melhora a política global.

Os gráficos da figura 5.55 ilustram a variação das políticas do *Ant-Q* e do *SAnt-Q* nos episódios t_{30}, t_{50} e t_{100}. É possível observar que as oscilações das políticas com a rede de relacionamentos diminuem quando comparadas com as imagens da figura 5.52.

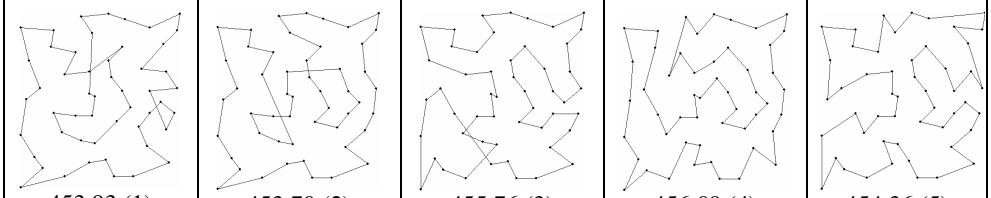


a. eil51 | b. eil76

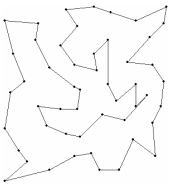
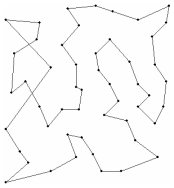
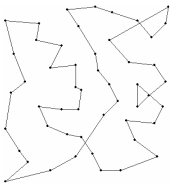
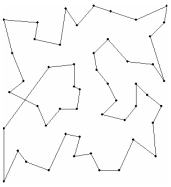
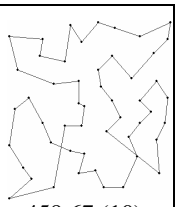
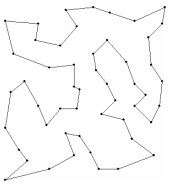
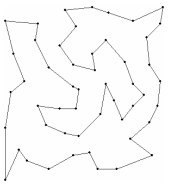
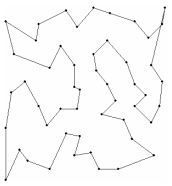
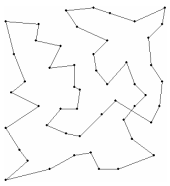
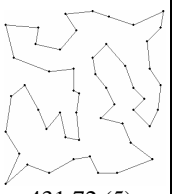
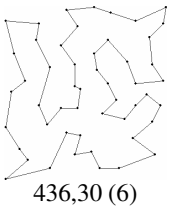
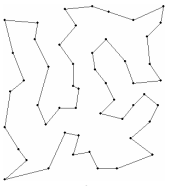
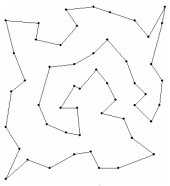
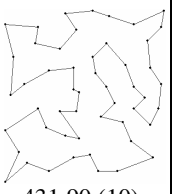
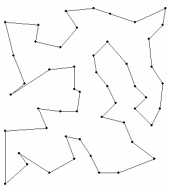
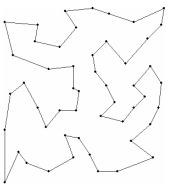
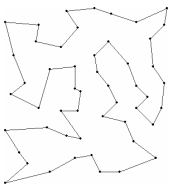
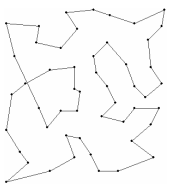
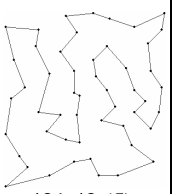
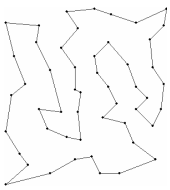
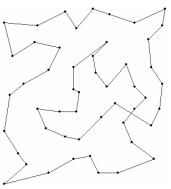
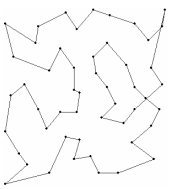
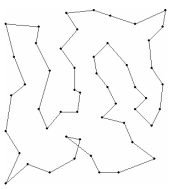
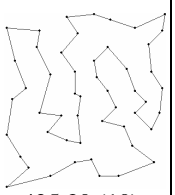
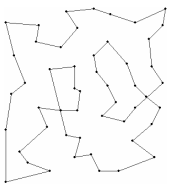
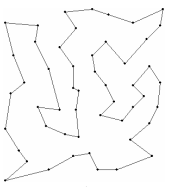
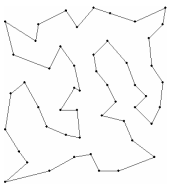
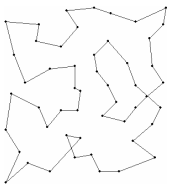
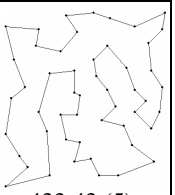
Figura 5.55: Oscilação das políticas com o *SAnt* com q_0=1 após os episódios t_{30}, t_{50} e t_{100}

As imagens das figuras 5.56 e 5.57 ilustram as melhores políticas descobertas para cada experimento, onde é possível observar que as políticas com o *SAnt-Q* apresentam as melhores soluções.

a. Soluções com o *Ant-Q* (eil51)



453,93 (1) | 453,70 (2) | 455,76 (3) | 456,09 (4) | 454,36 (5)

 451,65 (6)	 455,24 (7)	 455,39 (8)	 460,94 (9)	 459,67 (10)
b. Soluções com o *SAnt-Q* (eil51)				
 431,61 (1)	 438,99 (2)	 443,53 (3)	 440,51 (4)	 431,72 (5)
 436,30 (6)	 434,62 (7)	437,31 (8)	 440,86 (9)	 431,90 (10)
t_{30}				
 441,25 (1)	 439,89 (2)	 437,79 (3)	 439,48 (4)	 431,42 (5)
 436,64 (6)	 441,93 (7)	 445,28 (8)	 438,26 (9)	 435,20 (10)
t_{50}				
 451,98 (1)	 438,57 (2)	 437,69 (3)	 445,30 (4)	 433,43 (5)

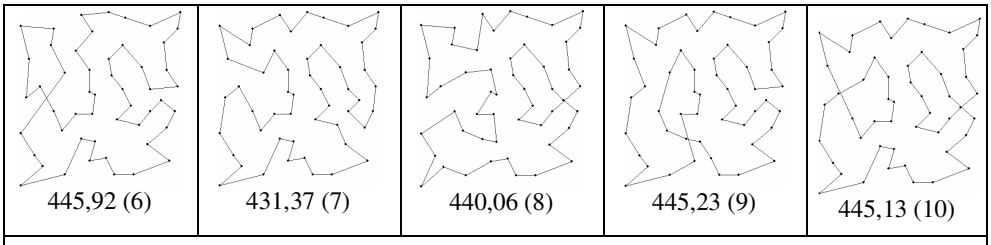
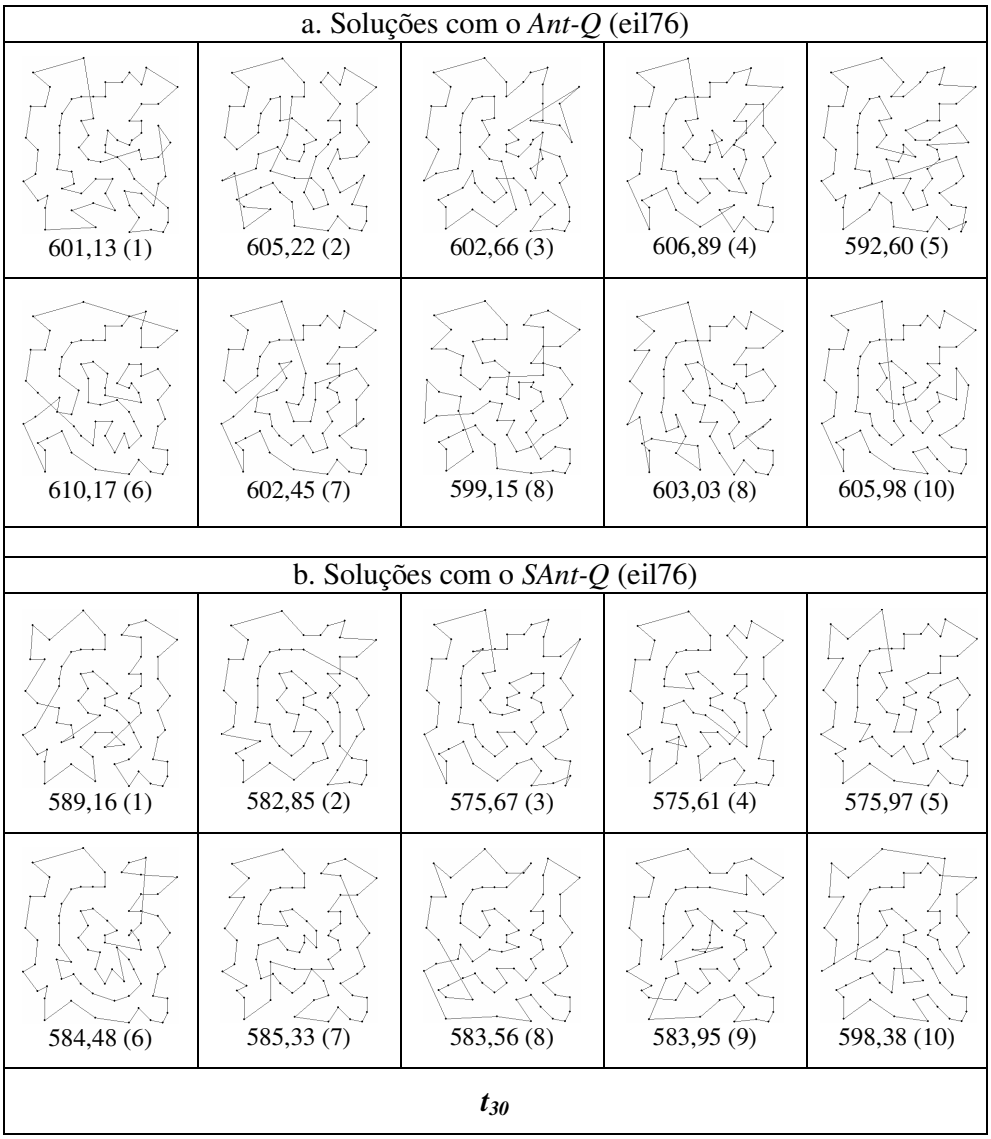


Figura 5.56: Soluções com o *Ant-Q* e *SAnt-Q* (eil51)



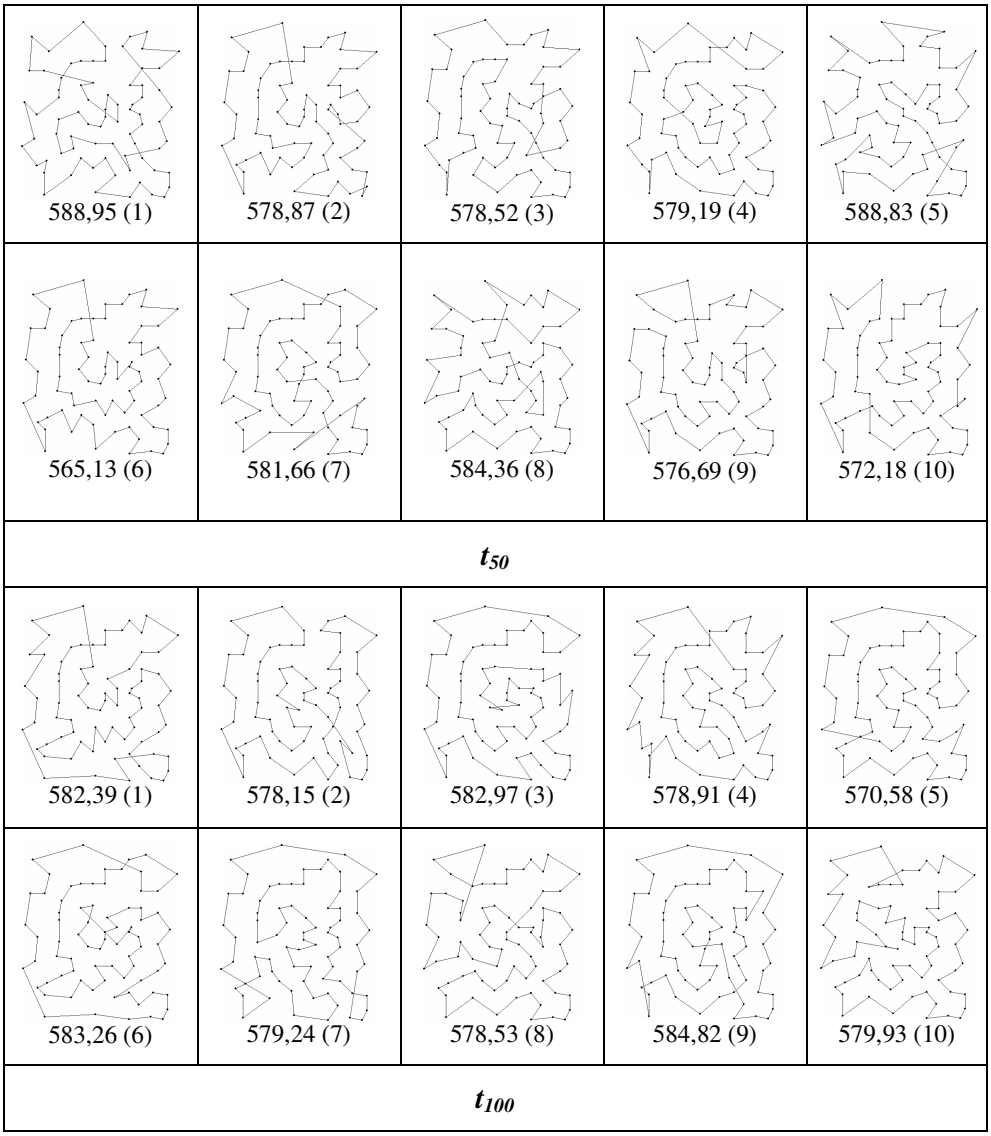


Figura 5.57: Soluções com o *Ant-Q* e *SAnt-Q* (eil76)

Como mencionado, foi usado o teste não-paramétrico de Friedman para verificar se há diferença significativa entre as políticas do *Ant-Q* e do *SAnt-Q*. Caso a hipótese nula seja caracterizada, as políticas dos algoritmos são equivalentes, uma vez que possuem ranques iguais. O resultado numérico do teste estatístico de Friedman utiliza um nível de significância (*p-valor*), caso este seja menor que 0,05, então é recomendado rejeitar a hipótese nula.

Foram consideradas as seguintes comparações nos problemas eil51 e eil76: *Ant-Q vs. SAnt-Q* t_{30}; *Ant-Q vs. SAnt-Q* t_{50}; e *Ant-Q vs. SAnt-Q* t_{100}. A tabela 5.13 mostra os *p-valor* obtidos com os conjuntos de instâncias eil51 e eil76.

Tabela 5.13: *p-valor* com o teste de Friedman

Episódio	eil51	eil76
t_{30}	p= 0,00026	p= 0,00078
t_{50}	p= 0,00026	p= 0,00078
t_{100}	p= 0,00026	p= 0,00078

Assim, para um valor de *p-valor* < 0,05 é possível concluir que existe diferença significativa entre os algoritmos, ou seja, a hipótese nula é rejeitada. Com isso, a confiança de haver diferença significativa entre o *Ant-Q* e *SAnt-Q* é de 95%. A tabela 5.14 mostra os resultados obtidos com os algoritmos, onde é mostrado o custo médio das políticas, a média e a soma dos ranques.

Tabela 5.14: Comparativo das médias com o teste de Friedman (500 episódios)

	Ant-Q		*SAnt-Q*					
			iniciando em t_{30}		iniciando em t_{50}		iniciando em t_{100}	
	eil51	eil76	eil51	eil76	eil51	eil76	eil51	eil76
Custo médio das políticas	455,67 (±4,768)	602,92 (±4,798)	436,73 (±4,246)	583,49 (±6,981)	438,71 (±3,840)	579,43 (±6,257)	441,46 (±6,357)	579,87 (±4,004)
Média dos ranques	2,000	2,000	1,000	1,000	1,000	1,000	1,000	1,000
Soma dos ranques	20,00	20,00	10,00	10,00	10,00	10,00	10,00	10,00

5.3.4 Método de Otimização Social

Foi observado na subseção 5.3.3 que a rede gerada seguindo alguns princípios de redes sociais é capaz de melhorar a utilidade das políticas geradas por algoritmos de colônia de formigas. No intuito de observar a evolução da rede em outras situações, o *SAnt-Q* foi testado sem a influência de algoritmos que utilizam funções heurísticas como a distância euclidiana, ou seja, tentamos verificar se a rede de relacionamentos gerada é capaz de garantir convergência para políticas de boa qualidade mesmo quando um algoritmo gerador produz políticas aleatoriamente. Dessa forma podemos caracterizar a independência da abordagem em relação ao algoritmo de otimização de entrada gerador de soluções candidatas.

Para auxiliar nessa abordagem, é incluído na equação 5.11 um parâmetro (ω) para privilegiar as relações entre estados mais centrais, que determina uma recompensa em função da centralidade de grau. Em redes com muitos agentes, os estados tendem a ter alto grau, devido à quantidade de interações dos agentes ao longo da aprendizagem. Quando isso ocorre, normalmente há atraso na aprendizagem, devido ao aumento de tempo para ligar os estados à

solução. Para aproveitar essa característica, é usada na rede de relacionamentos uma medida de centralidade para melhorar o crescimento da rede.

Nós adaptamos a abordagem de (Barabási *et al.* 2000), considerando que o crescimento da rede de relacionamentos se dá preferencialmente pelas relações de adjacências (díades) que ocorrem com maior frequência durante as gerações de políticas candidatas intermediárias no episódio atual. Portanto, a frequência das relações de estados díades é utilizada como fator preferencial para adição de novas relações (e possivelmente novos estados) na rede. Assim, o parâmetro de reforço $\omega(r_{s,u})$ é baseado na quantidade de vezes que a adjacência entre dois estados ocorreu dentre as políticas candidatas do episódio atual, *i.e.*, o número de vezes que os agentes (formigas) se moveram do estado *s* para o estado *u*. Essa abordagem permite afirmar que relações mais frequentes tendem a se repetir nas políticas, indicando convergência do algoritmo. Relações com baixa frequência possuem pouca influência na política. A equação 3.6 foi adaptada para a equação 5.17 para computar a frequência das interações dos estados:

$$\omega(r_{s,u})_t = \frac{\sum_{i=1}^{n} \begin{matrix} 1 & se \quad r_{s,u} \in AQ_{i_t} \\ 0 & se \quad r_{s,u} \notin AQ_{i_t} \end{matrix}}{n-1} \tag{5.17}$$

$$Q(r_{s,u})_t = (Q(r_{s,u})_{t-1} + [\inf(r_{s,u}) \times \rho] \times \omega) \qquad \text{Equação 5.11 modificada}$$

onde $\omega(r_{s,u})_t$ é a frequência das interações no episódio *t* e *n* é o número de políticas geradas (equivalente ao número de agentes e estados).

Outras métricas como centralidade por aproximação, intermediação e distância poderiam ser usadas e/ou adaptadas para privilegiar determinadas relações da rede de relacionamentos. Apesar dessas métricas, a centralidade de grau parece adequada devido às características do domínio e a simplicidade da adaptação ao estudo de caso deste trabalho.

Para substituir as políticas geradas com o *Ant-Q*, foi usado o *framework* apresentado na seção 5.2 para gerar políticas candidatas aleatoriamente para os problemas eil51 e eil76. As imagens das figuras 5.58 e 5.59 ilustram quinze políticas obtidas com o gerador de teste.

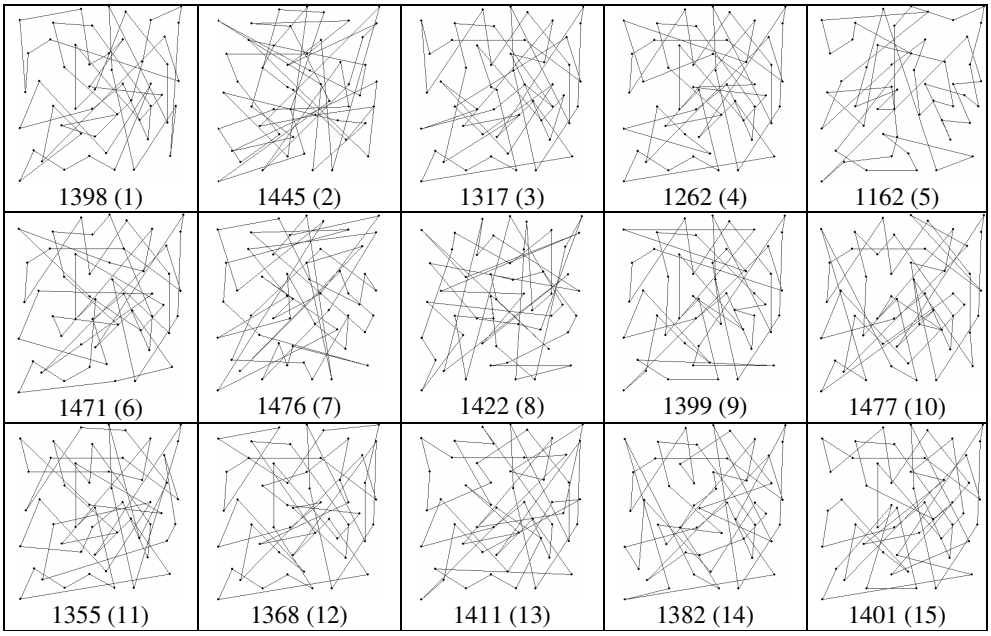


Figura 5.58: Políticas obtidas com o gerador de teste (eil51)

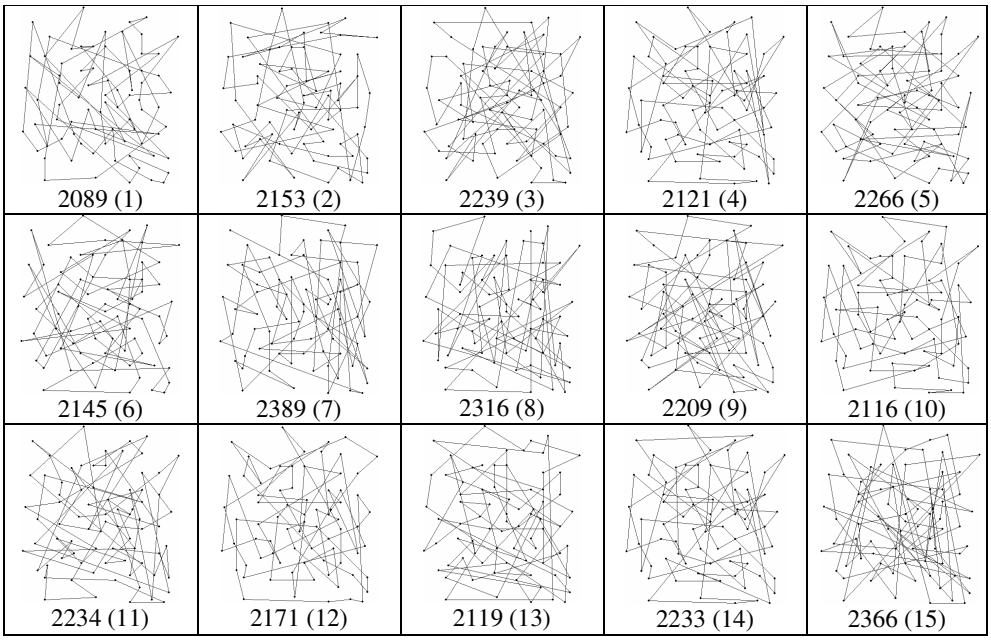
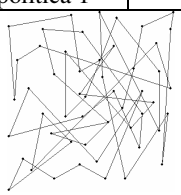
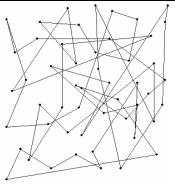
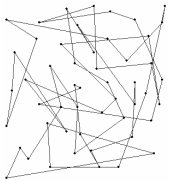
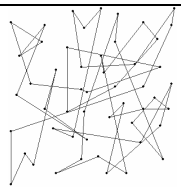
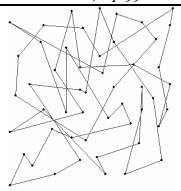
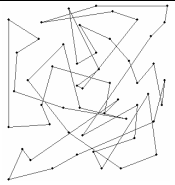
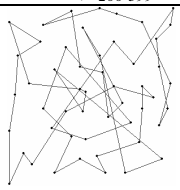
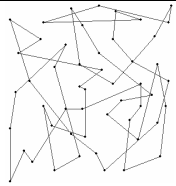
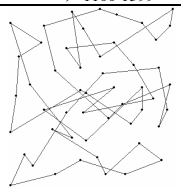
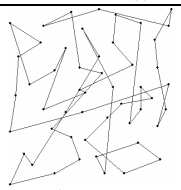
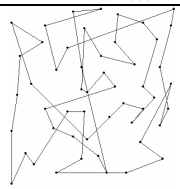
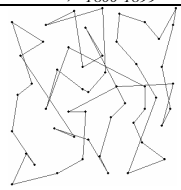
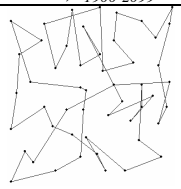
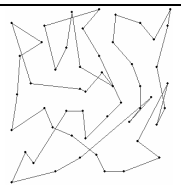
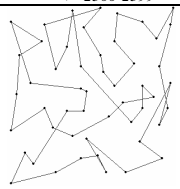
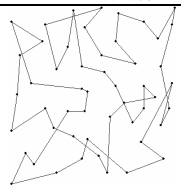
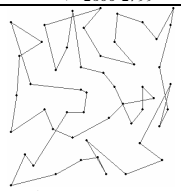
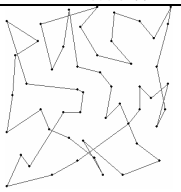
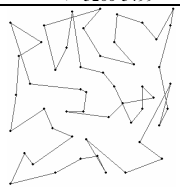
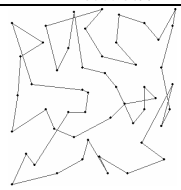
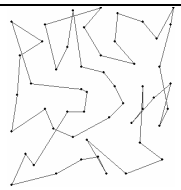
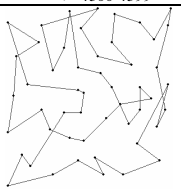
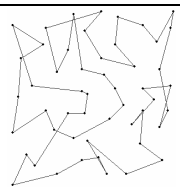
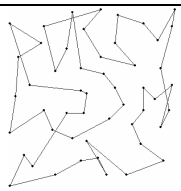
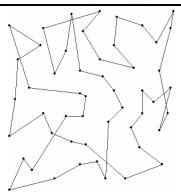
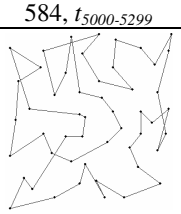
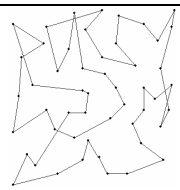
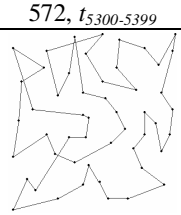


Figura 5.59: Políticas obtidas com o gerador de teste (eil76)

O procedimento para o crescimento da rede com as políticas do gerador de teste é o mesmo apresentado nos experimentos anteriores. Devido a quantidade de episódios utilizados nesses experimentos, foi adotado na regra de transição do *SAnt-Q* o parâmetro q_0 com valor 0. Usando os procedimentos apresentados na subseção 5.3.2 é ilustrada nas figuras 5.60 e 5.62 a evolução da rede, onde é mostrado o custo da política do *SAnt-Q* em intervalos de tempo. Os resultados foram obtidos usando 10000 episódios com os problemas eil51 e eil76.

Início (eil51), política 1

 1398, $t_{1\text{-}99}$	(1) ▶	 1103, $t_{100\text{-}199}$	(2) ▶	 1051, $t_{200\text{-}599}$	(3) ▶	 994, $t_{600\text{-}1099}$
 886, $t_{1100\text{-}1399}$	(4) ▶	 869, $t_{1400\text{-}1599}$	(5) ▶	 802, $t_{1600\text{-}1799}$	(6) ▶	 795, $t_{1800\text{-}1899}$
 753, $t_{1900\text{-}2099}$	(7) ▶	 746, $t_{2100\text{-}2299}$	(8) ▶	 709, $t_{2300\text{-}2399}$	(9) ▶	 697, $t_{2400\text{-}2599}$
 671, $t_{2600\text{-}2799}$	(10) ▶	 623, $t_{2800\text{-}3199}$	(11) ▶	 608, $t_{3200\text{-}3499}$	(12) ▶	 603, $t_{3500\text{-}3899}$
 599, $t_{3900\text{-}4299}$	(13) ▶	 597, $t_{4300\text{-}4399}$	(14) ▶	 594, $t_{4400\text{-}4499}$	(15) ▶	 593, $t_{4500\text{-}4899}$
 591, $t_{4900\text{-}4999}$	(16) ▶	 584, $t_{5000\text{-}5299}$	(17) ▶	 583, $t_{5300\text{-}5299}$	(18) ▶	 572, $t_{5300\text{-}5399}$
 568, $t_{5400\text{-}5599}$	(19) ▶	 566, $t_{5600\text{-}5699}$	(20) ▶	 555, $t_{5700\text{-}7000}$	(21) ▶	 531, $t_{7000\text{-}7399}$

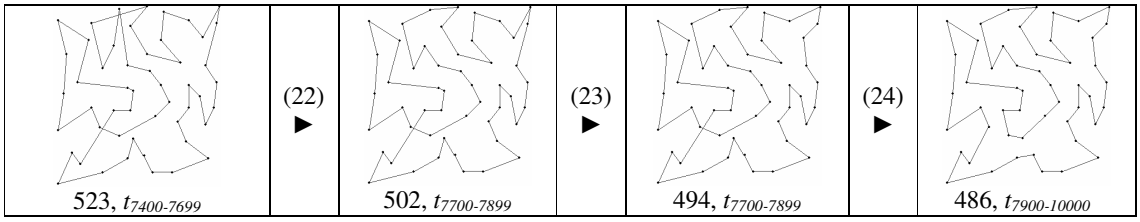


Figura 5.60: Evolução da rede com o método de otimização social (eil51)

As figuras 5.61 e 5.63 mostram as melhores políticas obtidas com o método de otimização social no problema eil51 para cada experimento.

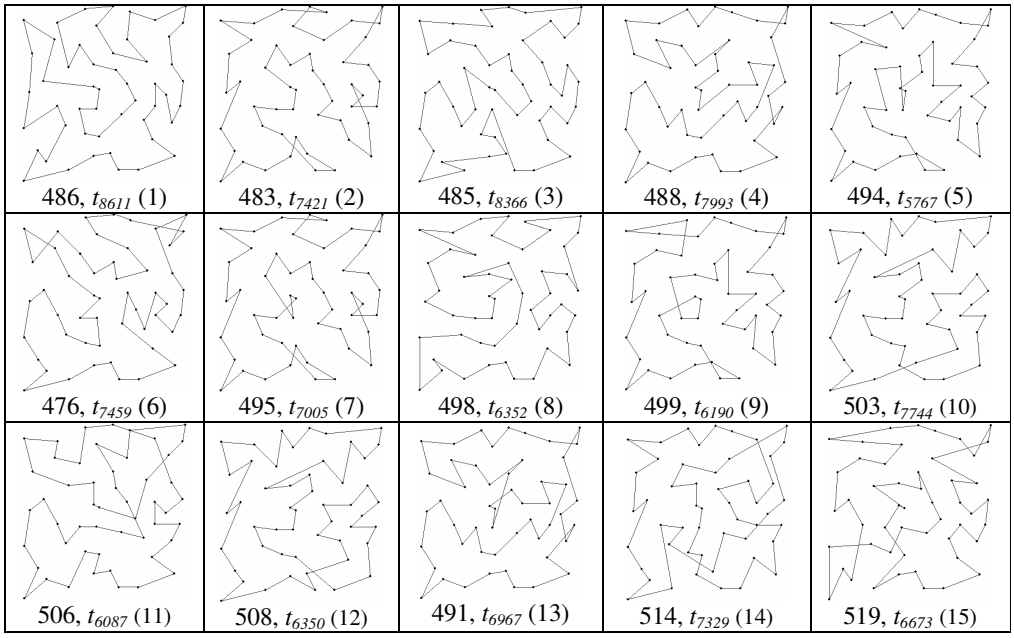
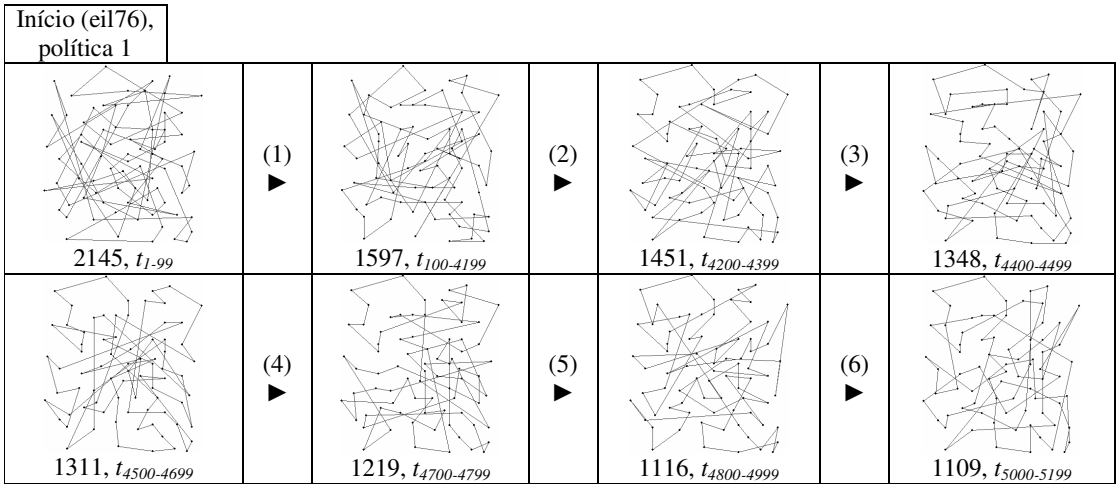


Figura 5.61: Políticas obtidas com o método de otimização social (eil51) em 10.000 episódios



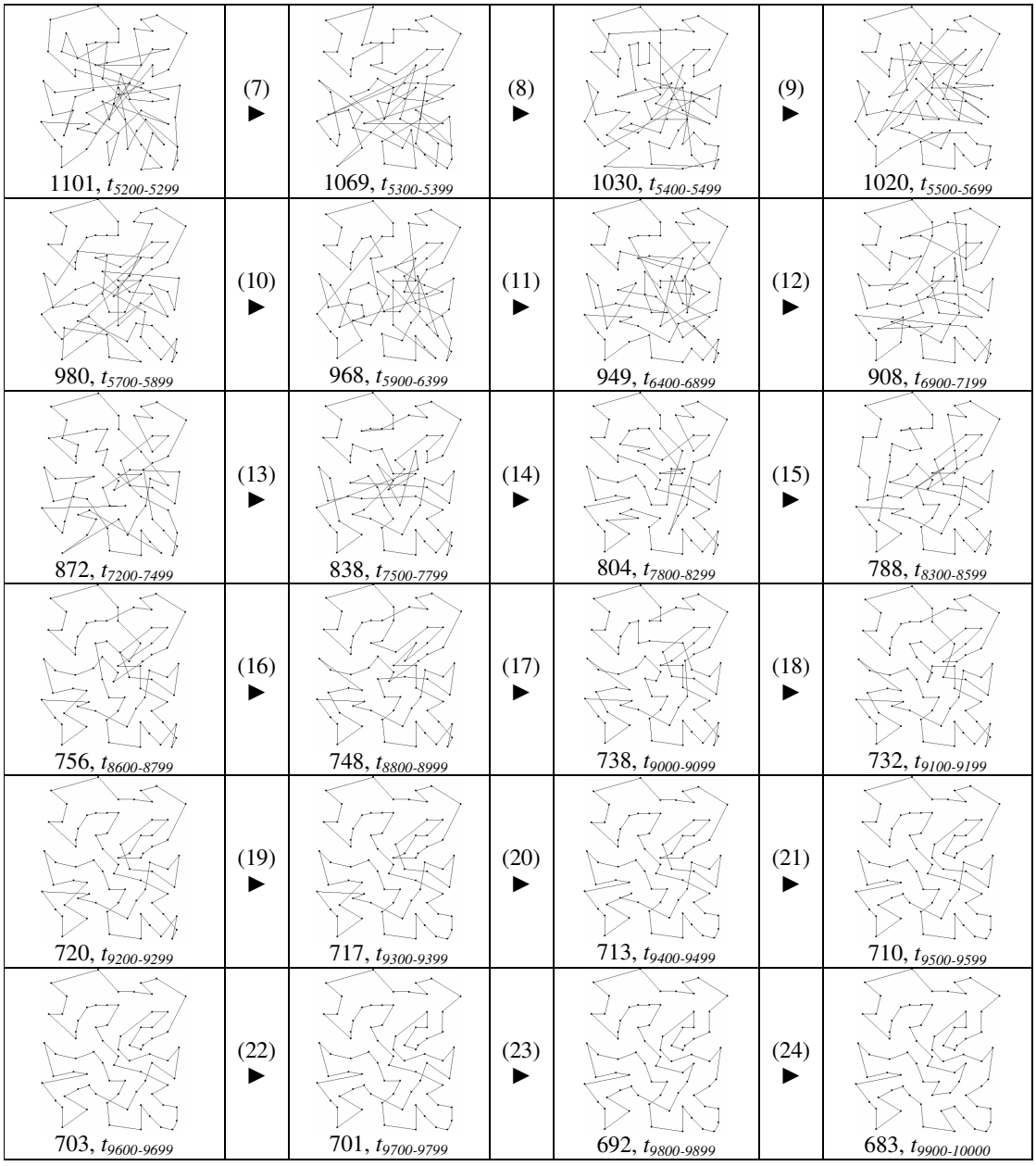


Figura 5.62: Evolução da rede com o método de otimização social (eil76) em 10.000 episódios

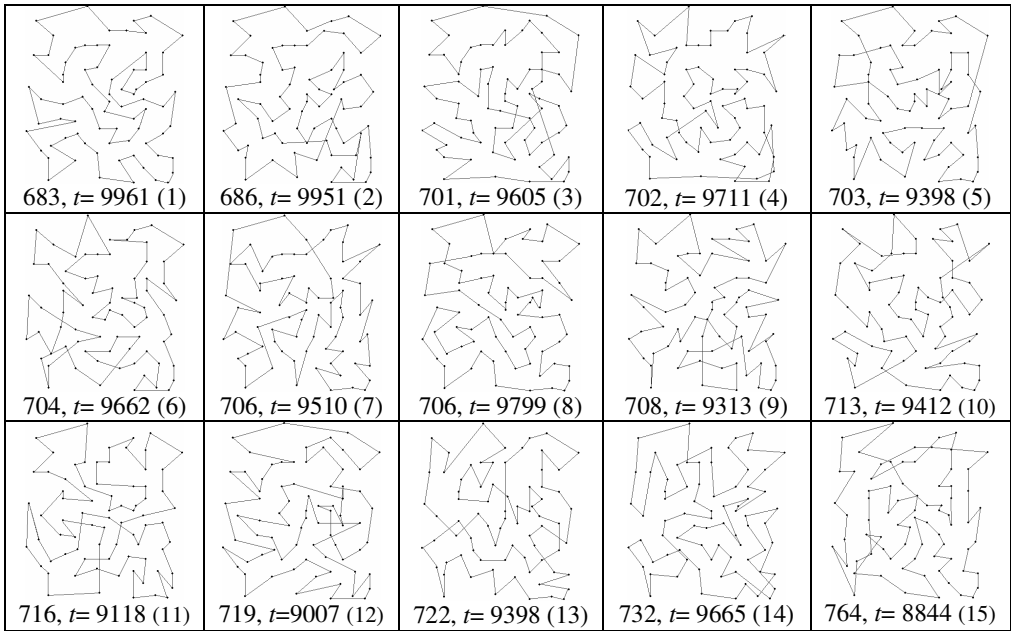


Figura 5.63: Políticas obtidas com o método de otimização social (eil76) em 10.000 episódios

É possível observar na figura 5.60 que durante a evolução da rede no problema eil51, as melhores políticas emergem a partir do episódio 3000. No problema eil76 (figura 5.62), as melhores políticas são encontradas a partir do episódio 9000, pois devido aos baixos valores das relações, o parâmetro de exploração q_0= 0 precisa de mais episódios para estabelecer as melhores relações em ambientes com muitos estados. O parâmetro de exploração q_0= 1 induz a política para uma convergência acelerada, no entanto com valores não satisfatórios, devido à estagnação em mínimos locais nos episódios iniciais.

Para comparar o método de otimização social com abordagens baseadas em recompensas que não utilizam heurísticas do domínio, foi utilizado o *Ant-Q* com o parâmetro heurístico β= 0, sendo a configuração dos demais parâmetros o mesmo dos experimentos anteriores. Assim, o *Ant-Q* com essa parametrização utiliza somente as recompensas adquiridas para guiar a busca no espaço de estados e usa as regras de atualizações (local e global) para estimar as recompensas. A tabela 5.15 mostra o custo médio dos valores das melhores políticas do método de otimização social e do *Ant-Q* sem heurística.

Tabela 5.15: Custo médio das políticas do *Ant-Q* sem heurística e do método social (eil51 e eil76)

	Ant-Q β= 0	Método Social
eil51	1146,33	495,46
eil76	1912,33	711

É possível observar na tabela 5.15 que o *Ant-Q* sem heurística (β= 0) não consegue a convergência para boas soluções usando somente as recompensas como fator de exploração, mesmo que haja muitos episódios e elevada quantidade de interações entre os agentes (díade). Essa observação é encontrada no método de otimização social, que aproveita a sociabilidade dos agentes com algoritmos por recompensas para melhorar o processo de tomada de decisão. Os gráficos das figuras 5.64 e 5.65 mostram os valores das políticas em cada experimento.

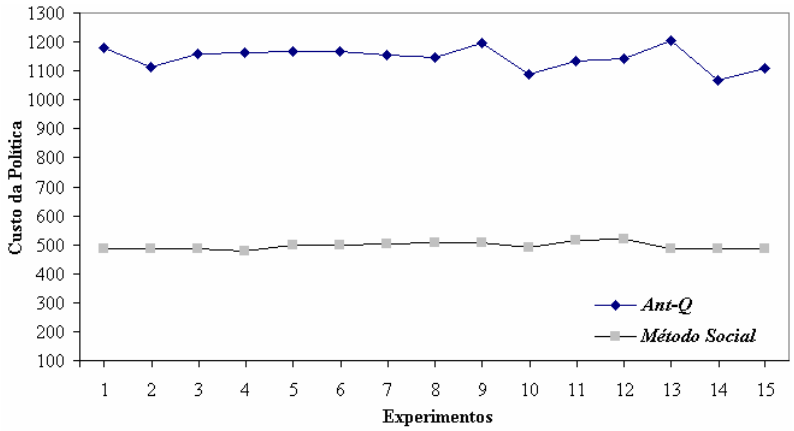


Figura 5.64: *Ant-Q* sem heurística *vs.* método social, eil51 com 10000 episódios

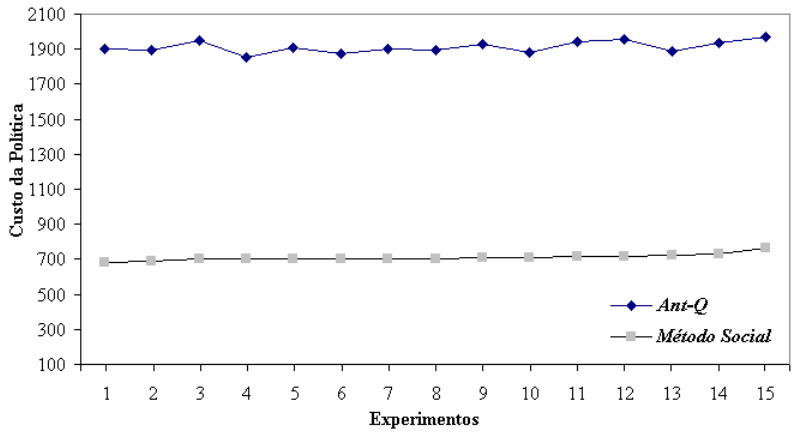


Figura 5.65: *Ant-Q* sem heurística *vs.* Método Social, eil76 com 10000 episódios

5.3.4.1 Discussões sobre o *SAnt-Q*

Os resultados experimentais mostram que mesmo sem um gerador de soluções heurístico a convergência ainda é possível com o método de otimização social *SAnt-Q*. As boas políticas (figuras 5.61 e 5.63) obtidas com esse método são decorrentes dos ajustes dos valores das intensidades das relações construídas pelos agentes durante a interação.

Isso demonstra que técnicas da análise de redes sociais podem melhorar algoritmos baseados em reforços que tem a sociabilidade como uma das principais características. A adaptação de uma métrica das redes sociais permitiu privilegiar as relações mais frequêntes na rede, sendo capaz de produzir forte influência nas relações utilizando aprendizagem por reforço e técnicas da análise das redes sociais. Isso foi demonstrado com o uso do parâmetro ω, que atua como uma alternativa às heurísticas baseadas em dados do domínio, beneficiando as relações mais frequentes.

Os resultados obtidos com o método de otimização social mostram que a sociabilidade decorrente das interações sociais, melhora a coordenação dos agentes com a estrutura social para a tomada de decisão. O comportamento coletivo a partir das interações é capaz de gerar um mecanismo de coordenação com comportamentos autônomos e locais, sem necessitar da coordenação centralizada.

A estrutura social construída a partir das interações desses indivíduos pôde melhorar a coordenação para o objetivo global. Como em outros sistemas sociais, o fortalecimento das relações entre indivíduos pares ou em grupos ocorre por algum tipo de relação, *e.g.*, trabalho, amizade, ou por proximidade geográfica ou tarefa comum, que ao longo do tempo são intensificadas e enfraquecidas. O método proposto segue esses princípios, e mostra que atitudes comportamentais individuais e coletivas podem ser estendidas para modelos computacionais.

Considerações finais

Neste capítulo foi apresentada a metodologia para aperfeiçoar métodos de coordenação empregando técnicas de redes sociais, aprendizagem por reforço e algoritmos de colônia de formigas. Inicialmente, discutimos como os agentes podem se coordenar compartilhando recompensas para alcançarem o objetivo estabelecido. Os resultados experimentais foram discutidos, analisando o impacto das recompensas compartilhadas no aprendizado dos agentes.

Foi observado que na aprendizagem por reforço a interação com modelos sociais que compartilham as recompensas pode em algum momento não satisfazer a política, pois a troca de conhecimento entre os agentes pode gerar novas políticas intermediárias incompatíveis com uma rápida convergência.

Para resolver esse problema, um método híbrido de aprendizagem foi implementado, sendo os resultados comparados com os modelos que não partilham recompensas. Foi mostrado o desenvolvimento de *frameworks* de testes, que permitiram analisar o impacto dos parâmetros dos algoritmos *Ant-Q* e *Q-learning* na convergência do sistema. A última parte da me-

todologia abordou como a sociabilidade dos indivíduos que utilizam algoritmos de colônia de formigas é importante para melhorar o comportamento coletivo na tomada de decisões.

Os resultados mostram que as técnicas da análise das redes sociais são úteis na formalização do processo para a construção de estruturas sociais. A melhora da coordenação de agentes baseado nessas discussões comprova o aspecto inovador desde trabalho, mostrando que as pesquisas apresentadas a partir da adaptação de métodos de coordenação multiagente, colônia de formigas, aprendizagem por reforço e redes sociais indicam a comprovação das hipóteses apresentadas na seção 1.2.

Capítulo 6

Conclusões e Discussões Finais

O comportamento coletivo quando coordenado atribui aos indivíduos de um sistema habilidades (recompensas pelas atitudes) e padrões de comportamentos que melhoram a interação. Em um sistema com características sociais a coordenação entre os indivíduos é necessária, pois a troca de informações deve beneficiar tanto o comportamento coletivo como o individual. O comportamento é social quando dois ou mais indivíduos dependem mutuamente um do outro para a execução de tarefas em um ambiente social.

Neste trabalho defendemos a ideia que a partir da integração e adaptação de métodos de coordenação multiagente, colônia de formigas, aprendizagem por reforço e redes sociais os indivíduos pudessem socializar as informações interagindo em uma estrutura social dinâmica para melhorar a coordenação. O texto em destaque nos próximos parágrafos está relacionado às respostas dos questionamentos da seção 1.2, mostrando a comprovação da hipótese principal deste trabalho. Através dos conceitos da teoria e análise das redes sociais e da estrutura social construída a partir das interações usando algoritmos de aprendizagem por reforço foi possível alcançar e satisfazer os objetivos principais assumidos.

Observamos que as **interações sociais** estabelecem relações entre os indivíduos de um sistema multiagente que realizam alguma atividade em comum a partir da tomada de decisão coletiva. Neste contexto, um indivíduo atua em um ambiente estabelecendo ligações com os demais indivíduos a partir das recompensas geradas por algoritmos de aprendizagem por reforço. Esses indivíduos socializam as recompensas para melhorar o seu comportamento e possivelmente de outros agentes, caracterizando a interação social.

Para que a interação social tenha efeito sobre os indivíduos, é necessário que ela produza alterações no comportamento individual (Becker, 1974). Portanto, a interação social é o

resultado do comportamento coletivo dos indivíduos que utilizam recompensas sociais para a manutenção de suas relações e vice-versa.

É possível verificar na metodologia proposta que a interação social é fundamental para o desenvolvimento de comportamentos coletivos, pois atua na construção da **estrutura social** e na qualidade da coordenação. Ressalta-se também a importância dos princípios da sociabilidade nas atividades em comum dos indivíduos. Muitas vezes esses conceitos estão relacionados a teorias de ação e modelos de sistemas sociais, onde as interações entre os indivíduos devem resultar dos modelos de aprendizagem construídos a partir da própria interação, levando em consideração as mudanças que ocorrem entre os membros do grupo, aperfeiçoando o desempenho das tarefas.

Embora as características sociais de um sistema sejam reconhecidamente importantes para a coerência de comportamentos dos indivíduos, destaca-se também que a **formalização** dessas **influências** constitui uma área de pesquisa ainda pouco explorada. Neste sentido, este trabalho defende a utilização de medidas de centralidade e intensidade das relações entre pares de indivíduos (díades) oriundas da análise das redes sociais. A **adaptação** e a **construção da rede de relacionamentos** ocorrem segundo modelos e equações matemáticas, que modelam a influência dos indivíduos. Para melhorar a adaptação da rede de relacionamentos, devem ser utilizadas as melhores políticas identificadas. O **crescimento da rede** se dá preferencialmente pelas relações das díades mais influentes, que ocorrem com maior frequência durante as gerações de políticas candidatas. Isso é determinado pela sobreposição dos indivíduos com as melhores políticas do sistema, onde os indivíduos com maior força influenciam os demais através de recompensas individuais.

A **identificação** dos **indivíduos mais relevantes** pode ser realizada observando aqueles cujos comportamentos são reproduzidos por outros indivíduos intensificando as relações entre estados e ações específicos. Neste caso, as medidas de centralidade da análise das redes sociais auxiliam na determinação de comportamentos desejáveis, produzindo melhores políticas coletivas de ação.

Quando políticas de ação devem ser construídas em um ambiente coletivo, modelos específicos de geração e **compartilhamento de recompensas** devem ser empregados, como por exemplo, compartilhando recompensas (i) em episódios pré-determinados, (ii) a cada ação, a partir de uma regra de transição baseada na própria política de ação, e (iii) de forma local e global. Esses modelos incluem no processo de geração de comportamentos uma **dimensão social** implícita, mas que também pode ser constituída por uma estrutura social explicitamente representada.

Uma **estrutura social** pode ser gerada, por exemplo, a partir de uma rede de feromônio produzida por um algoritmo de colônia de formigas. A partir do comportamento dos indivíduos e da aplicação da teoria da análise de redes sociais, é possível identificar a estrutura social (rede) e padrões de comportamentos entre os estados do sistema, destacando quem interage com quem, a frequência e a intensidade de interação. O conhecimento adquirido pelos agentes permite que as relações mais intensas tenham maior probabilidade de serem incluídas na geração da rede, favorecendo os estados proeminentes e diminuindo consequentemente a intensidade das relações potencialmente inúteis.

É possível mostrar as principais contribuições do algoritmo *SAnt-Q* a partir dos resultados apresentados. A estrutura social gerada **melhora o comportamento** do *Ant-Q*, **identificando a topologia** que emerge a partir de determinados episódios, produzindo uma rede de relacionamentos entre os estados do sistema. A topologia inicial da rede gerada com o algoritmo *SAnt-Q* é semelhante ao modelo de *redes* e *grafos aleatórios*. Essa topologia de rede não é a desejada no início da aprendizagem, pois o grau da maioria dos estados é semelhante, o que ocasiona mesma probabilidade de se conectarem aos demais estados. Entretanto, iterativamente, as relações da rede de relacionamentos são intensificadas, alterando a topologia para uma rede do tipo *mundo pequeno*.

A alteração das características da rede ocorre porque algoritmos de colônia de formigas são dotados de mecanismos que induzem os agentes a usarem estratégias exploratórias, devido ao parâmetro que define a taxa de exploração e induz os agentes a ações baseadas em probabilidade. Estados com recompensas menores também podem ser escolhidos, na intenção de maximizar as recompensas no final da aprendizagem. De uma forma geral, estados mais próximos tendem a estar conectados com mais intensidade. Porém, alguns estados estarão relacionados com estados mais distantes, criando novas conexões e reduzindo o tamanho médio do caminho entre todos os estados, favorecendo a descoberta de boas soluções.

6.1 Trabalhos Futuros

Os resultados observados com os modelos de compartilhamento de recompensas sociais indicam que novas pesquisas podem ser realizadas em ambientes mais complexos e com elevada dinamicidade. Alguns experimentos preliminares neste tipo de ambiente mostram que os modelos desenvolvidos podem favorecer a convergência para políticas de boa qualidade neste tipo de problema (seção 5.1).

Apesar dos resultados obtidos com os modelos sociais serem encorajadores, algumas diretivas futuras são merecedoras de investigações, como por exemplo, avaliar a interação dos agentes quando compartilham recompensas em ambientes com centenas de agentes. Como os agentes interagem compartilhando recompensas de modelos diferentes, *a priori*, os agentes deverão ter um comportamento coletivo adequado, pois as recompensas obtidas com o modelo híbrido parece menos susceptível a ruídos nos dados de aprendizagem. Portanto, esses estudos complementares são importantes para a evolução dos modelos sociais, pois a diversidade de experimentos a serem produzidos é enorme, o que faz surgir novos cenários a serem estudados.

Apesar dos resultados encorajadores com o *SAnt-Q*, é possível que novas estratégias de atualização, como por exemplo, procedimentos para atualizações globais, possam melhorar a coordenação dos agentes, pois a intensidade de algumas relações pode levar a mínimos locais. Outra alternativa é o uso de uma estratégia para melhorar a busca global, onde procedimentos poderiam penalizar as relações com valores muito elevados. Isso poderia ser aplicado a um número específico de episódios, diminuindo o valor de algumas relações favorecendo o surgimento de novos relacionamentos (novas soluções de boa qualidade).

Neste trabalho foi observado que a rede de relacionamentos entre estados do sistema, gerada a partir dos comportamentos dos agentes (recompensas) e valores em função de uma centralidade de grau, produziram bons resultados em problemas de *benchmark.* Mesmo desconsiderando o uso de funções heurísticas específicas do domínio (*e.g.* distância geográfica entre estados) a topologia da rede de relacionamentos é suficiente para gerar novos comportamentos. Outra alternativa baseada em uma função de centralidade seria utilizar outras métricas da análise das redes sociais, como por exemplo, medidas de centralidade por aproximação e intermediação, onde poderiam privilegiar determinadas relações da rede de relacionamentos. Essas medidas poderiam ser usadas conforme a manutenção da díade e dos valores utilizados para o parâmetro de aprendizagem ρ (equação 5.12).

6.2 Publicações Relacionadas

Nesse trabalho procurou-se desenvolver e melhorar metodologias científicas voltadas para duas direções principais: (i) o desenvolvimento de modelos para o compartilhamento de recompensas sociais e (ii) a utilização da estrutura social construída com a sociabilidade para melhorar o comportamento dos indivíduos de um sistema multiagente. Esses objetivos foram alcançados com as metodologias apresentadas no capítulo 5, onde as principais inovações

apresentadas permitiram a adaptação de métodos de coordenação multiagente, colônia de formigas e aprendizagem por reforço a partir de conceitos oriundos das redes sociais. Essas contribuições servirão como base para a elaboração de trabalhos científicos relacionados e também originaram alguns trabalhos já publicados listados a seguir:

- *Otimização dos Parâmetros de Aprendizagem para a Coordenação dos Agentes em Algoritmos de Enxames* (SCA - Simpósio de Computação Aplicada, Passo Fundo, 2009, ISNN 2176-8196 - Ribeiro, R.; Ronszcka, A. F.; Borges, A. P.; Enembreck, F.);
- *A Strategy for Converging Dynamic Action Policies* (IEEE Symposium Series on Computational Intelligence, 2009, Nashville. Proceedings of IEEE Symposium Series on Computational Intelligence, p. 136-143, March, 2009, ISBN 978-1-4244-2767-3 - Ribeiro, R.; Borges, A. P.; Koerich, A.; Scalabrin, E. E.; Enembreck, F.);
- *Uma Arquitetura de Aprendizagem para a Condução Automática de Veículos* (In. Proceedings of XXXV Conferencia Latinoamericana de Informática, 2009 - Borges, A. P.; Ribeiro, R.; Leite, A. R.; Dordal, O.; Giacomet, B.; Ávila, B. C.; Enembreck, F.; Scalabrin, E. E.);
- *A Learning Agent to Help Drive Vehicles* (In: 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), Santiago, 2009b, p. 282-287 - Borges, A. P.; Ribeiro, R.; Ávila, B.C.; Enembreck, F.; Scalabrin, E. E.)
- *Discovering Action Policies in Dynamic Environments* (Ágora (Caçador), v. 15, n.1 p. 175-185, 2008, ISSN 0104-7507 - Ribeiro, R.; Borges, A. P.; Ulbrich, G.; Koerich, A. L.; Scalabrin, E. E.; Enembreck, F.);
- *Interaction Models for Multiagent Reinforcement Learning* (International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA08, Vienna, Austria, 2008 - Ribeiro, R., Borges, A. P. e Enembreck, F.);
- *Noise Tolerance in Reinforcement Learning Algorithms* (IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT2007), Silicon Valley, California, USA. Proceedings of the IAT 2007. Los Alamitos: IEEE Computer Society, p. 265-268, 2007, ISSN/ISBN 0769530273 - Ribeiro, R.; Koerich A. L. and Enembreck F.);
- *Reinforcement Learning: Adaptive Agents for Discovery of Policies of Action* (Ágora (Caçador), v. 14, p. 9-24, 2007, ISSN 0104-7507 - Ribeiro, R.; Koerich, A. L.; Enembreck, F.).

Referências Bibliográficas

ABDALLAH, S.; LESSER, V. R. *Organization-based cooperative coalition formation*. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), 2004, p.162-168.

ABDALLAH, S.; LESSER, V. R. *Learning scalable coalition formation in an organizational context*. Coordination of Large-Scale Multiagent Systems, Springer, 2006, p. 195-215.

ABD-EL-BARR, M.; SAIT, S. M.; SARIF, B. A. B.; AL-SAIARI, U. *A modified ant colony algorithm for evolutionary design of digital circuits*. IEEE Congress on Evolutionary Computation (1) 2003, p. 708-715.

ANNALURU, R.; DAS, S.; PAHWA, A. *Multi-Level Ant Colony Algorithm for Optimal Placement of Capacitors in Distribution Systems*. IEEE Congress on Evolutionary Computation, 2004, p. 1932-1937.

ARAÚJO, R. M.; LAMB, L. C. *Memetic Networks: analyzing the effects of network properties in multi-agent performance*. In: Twenty-Third Conference on Artificial Intelligence (AAAI-08), 2008, Chicago. Menlo Park, CA: Association for the Advancement of Artificial Intelligence Press (AAAI Press), v. 1, 2008, p. 1-6.

ARKIN, R. *Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation*. Robotics and Autonomous Systems, Special Issue on Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. P. Maes, ed., v. 6, n. 1-2, 1990, p. 105-122.

BARABÁSI, A-L; ALBERT, R.; JEONG, H. *Scale-free characteristics of random networks: The topology of the World Wide Web*. Physical A, v. 281, 2000, p. 69-77.

BARABÁSI, A-L. *Linked, The new science of networks*. Perseus Publishing, Cambridge, Massachusetts, 2002.

BARABÁSI, A-L. *Linked. How Everything is Connected to Everything else and What it means for Business, Science and Every day Life*. Cambridge: Plume, 2003a.

BARABÁSI, A-L; BONABEAU, E. *Scale-Free Networks*. Scientific American, Issue 5, 60 (2003b), New York, NY, 2003, p. 60-69.

BASTOS FILHO, C. J. A.; LIMA NETO, F. B.; LINS, A. J. C. C.; NASCIMENTO, A. I. S.; LIMA, M. P. *A Novel Search Algorithm based on Fish School Behavior*. IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, p. 2646-2651.

BATAGELJ, V.; MRVAR, A. *Pajek - A program for large network analysis.* Lecture Notes in Computer Science, Graph Drawing, v. 2265, 2002, p. 8-11.

BATAGELJ, V.; MRVAR, A. *Pajek: Package for large networks*. Version 0.92. Ljubljana: University of Ljubljana, 2003a.

BATAGELJ, V.; MRVAR, A. *Pajek. Analysis and visualization of large networks*. n Junger, M., and Mutzel, P. (eds.), Graph Drawing Software. New York: Springer. 2003b, p. 77-44.

BEAN, N.; COSTA, A. *An analytic modeling approach for network routing algorithms that use ant-like mobile agents*. Computer Networks: The International Journal of Computer and Telecommunications Networking, v. 49, 2005, p. 243-268.

BECKER, G. S. *A Theory of Social Interactions*. University of Chicago Press. Journal of Political Economy, vol. 82, no. 6, 1974, p. 1063-1093.

BELL, J. E.; MCMULLEN, P. R. *Ant colony optimization techniques for the vehicle routing problem*. Advanced Engineering Informatics, v. 18, 2004, p. 41-48.

BENI, G., WANG, J. *Swarm Intelligence in Cellular Robotic Systems*. Proceeding NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June, 1989, p. 26-30.

BERTSEKAS, D. P. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, N.Y, 1987.

BIANCHI, L.; GAMBARDELLA, L. M.; DORIGO, M. *An Ant Colony Optimization Approach to the probabilistic Traveling Salesman Problem*, In Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature, LNCS. Springer Verlag, Berlin, Germany, 2002.

BOER, P.; HUISMAN, M.; SNIJDERS, T. A. B.; ZEGGELINK, E. P. H. *StOCNET: An open software system for the advanced statistical analysis of social networks.* Version 1.4. Groningen: ProGAMMA/ICS, University of Groningen. 2003.

BORGATTI, S. P. *NetDraw 1.0: Network visualization software*. Version 1.0.0.21. Harvard: Analytic Technologies, 2002a.

BORGATTI, S. P.; EVERETT, M. G.; FREEMAN, L. C. *UCINET 6 for Windows: Software for social network analysis*. Harvard: Analytic Technologies, 2002b.

BORGES, A. P.; RIBEIRO, R.; ÁVILA, B.C.; ENEMBRECK, F.; SCALABRIN, E. E. *A Learning Agent to Help Drive Vehicles*. In: 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), Santiago, 2009b, p. 282-287.

BORGES, A. P.; RIBEIRO, R.; LEITE, A. R.; DORDAL, O.; GIACOMET, B.; ÁVILA, B. C.; ENEMBRECK, F.; SCALABRIN, E. E. *Uma Arquitetura de Aprendizagem para a Condução Automática de Veículos*. In. Proceedings of XXXV Conferência LatinoAmericana de Informática, 2009c, p. 1-6.

BOWMAN, R. S.; HEXMOOR, H. *Agent collaboration and social networks*, Integration of Knowledge Intensive Multi-Agent Systems, April, 2005, p. 211-214.

BRADSHAW, J. M. An Introduction to software Agents. In: Bradshaw, J. M. (Ed.). Software Agents. Massachusetts: MIT Press 1997.

BROOKS, R. A. *A robust layered control system for a mobile robot*. IEEE J. Rob. Autom 2. 1986, p. 14-23.

BROOKS, R. A. *Elephants Don't Play Chess*, Robotics and Autonomous Systems, v. 6, 1990, p. 3-15.

BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C. *A new rank-based version of the Ant System: a computational study*. Central European Journal for Operations Research and Economics, v. 7(1), 1999a, p. 25-38.

BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C. *An improved ant system algorithm for the vehicle routing problem*. Annals of Operations Research, v.89, 1999b, p. 319-328.

BURT, R. S. *STRUCTURE.* Version 4.2. New York: Columbia University. 1991.

CASTELFRANCHI, C.; MICELI, M.; CESTA, A. *Dependence Relations among Autonomous Agents*, in Y.Demazeau, E.Werner (Eds), Decentralized A.I., Elsevier (North Holland), 1992.

CASTELFRANCHI, C. *To Be or Not To Be an Agent*, Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, p. 37-39.

CHAPELLE, J.; SIMONIN, O.; FERBER, J. *How Situated Agents can Learn to Cooperate by Monitoring their neighbors' Satisfaction*. ECAI'2002, Lyon, 2002, p. 68-72.

CHECHETKA, A.; SYCARA, K. *No-commitment branch and bound search for distributed constraint optimization*, Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, Hakodate, Japan, 2006, p. 1427-1429.

CHRISTOFIDES, N.; EILON, S. *An Algorithm for the Vehicle-Dispatching Problem.* Operations Research Quarterly 20, 1969, p. 309-318.

COELLO, C. A. C.; GUTIÉRREZ, R. L. Z; GARCÍA, B. M.; AGUIRRE, A. H. Automated Design of Combinational Logic Circuits using the Ant System. Engineering Optimization, v. 34, n. 2, 2002, p. 109-127.

COELLO, C. A. C.; TOSCANO, G.; LECHUGA, M. S. *Handling Multiple Objectives with Particle Swarm Optimization*. IEEE Transactions on Evolutionary Computation, v. 8, n. 3, 2004, p. 256-279.

CONWAY, L.; LESSER, V. R.; CORKILL, D. G. *The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks*. AI Magazine, 4(3):15-33, 1983.

COSTA, D; HERTZ, A. *Ants can colour graphs*, Journal of the Operational Research Society, v. 48, 1997, p. 295-305.

CRITES, R. H.; BARTO, A.G. *Improving Elevator Performance Using Reinforcement Learning*. Advances in Neural Information processing Systems 8. MIT Press, Cambridge, MA, 1996, p. 1017-1023.

CYRAM. *Cyram NetMiner II*. Version 2.0.5. Seoul: Cyram Co., Ltd. 2003.

DAUTENHAHN, K. *Getting to know each other - artificial social intelligence for autonomous robots*, Robotics and Autonomous Systems, v. 16, 1995, p. 333-356.

DAUTENHAHN, K.; CHRISTALLER, T. *Remembering, rehearsal and empathy - towards a social and embodied cognitive psychology for artifacts*. In Two Sciences of the Mind. Readings in cognitive science and consciousness, S. O'Nuallain and P. McKevitt, Eds. John Benjamins Publ., 1996, p. 257-282.

DeCANIO, S.; WATKINS, W. *Information processing and organizational structure*. Journal of Economic Behavior and Organization. V.36, pp. 275–294, 1998.

DECKER, K. S.; LESSER, V. R. *Generalizing the Partial Global Planning Algorithm*. International Journal on Intelligent Cooperative Information Systems, v. 1, n. 2, 1992, p. 319-346.

DECKER, K. S.; LESSER, V. R. *Quantitative modeling of complex computational task environments*. In Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, 1993, p. 217-224.

DECKER, K. S.; LESSER, V. R. *Designing a Family of Coordination Algorithms.* In Proceedings of the First International Conference on Multi-Agent Systems, AAAI Press: San Francisco, CA, San Francisco, 1995, p. 73-80.

DELOACH, S. A.; VALENZUELA, J. L. *An Agent-Environment Interaction Model*. In L. Padgham and F. Zambonelli (Eds.): AOSE 2006, LNCS 4405. Springer-Verlag, Berlin Heidelberg, 2007, p. 1-18.

DEMSAR, J. *Statistical Comparisons of Classifiers over Multiple Data Sets*. Journal of Machine Learning Research, 7: 1-30, 2006.

DI CARO, G.; DORIGO, M. *AntNet: Distributed stigmergetic control for communications networks*. Journal of Artificial Intelligence Research (JAIR), AI Access Foundation and Morgan Kaufmann Publishers, 1998, p. 317-365.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. *Positive feedback as a search strategy*. Technical Report TR91-016, Dip. Elettronica, Politecnico di Milano, Italy, 1991a.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. *The Ant System: an autocatalytic optimization process*. Technical Report TR91-016 Revised. Dipartimento di Elettronica, Politecnico di Milano, Itália, 1991b.

DORIGO, M. *Optimization, learning, and natural algorithms*. PhD thesis, Dip. Elettronica, Politecnico di Milano, Italy, 1992.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. *Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1): 1996, p. 29-41.

DORIGO, M.; GAMBARDELLA, L. M. *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*. IEEE Transactions on Evolutionary Computation, 1(1): 1997, p. 53-66.

DORIGO, M.; DI CARO, G.; GAMBARDELLA, L. M. *Ant algorithms for distributed discrete optimization*. Artificial Life, 5(2): 1999, p. 137-172.

DOYLE, J. *Rationality and its role in reasoning*. Computational Intelligence, v. 8, p. 376-409, 1992.

DURFEE, E. H. *Coordination of Distributed Problem Solvers*. Kluwer Academic Press, Boston, 1988.

DURFEE, E. H.; LESSER, V. R. *Partial Global Planning: A coordination framework for distributed hypothesis formation*. IEEE Transactions on Systems, Man, and Cybernetics, 21(5): 1991, p. 1167-1183.

DURFEE, E. H. *Planning in distributed artificial intelligence*. In: O'Hare, Greg; Jennings, Nick (Eds.). Foundations of distributed artificial intelligence, Willey, 1996.

DURFEE, E. H. *Distributed Problem Solving and Planning*. Chapter 3 in Gerhard Weiss, editor. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge MA, 1999.

EBERHART, R. C.; KENNEDY, J. F. *A new optimizer using particle swarm theory*. In: Proceedings of the sixth international symposium on micromachine and human science, Nagoya, Japan; 1995, p. 39-43.

ENEMBRECK, F. *Contribution à la conception d'agentes assistants personnels adaptatifs*, Thèse de Docteur. Université de Technologie de Compiègne U. F. R. de Sciences Et Technologie, 2003.

ENGELBRECHT, A. P. *Fundamentals of Computational Swarm* I*ntelligence*. Chichester: J. Wiley & Sons, 2005.

ERDÖS, P.; RÉNYI, M. *On Random Graphs*. Publication of the Mathematical Institute of The Hungarian Academy of Sciences, v. 5, 1960, p. 17-61.

FARATIN, P.; SIERRA, C.; JENNINGS, N. R. *Negotiation Decision Functions for Autonomous Agents*. Int. Journal of Robotics and Autonomous Systems, 24 (3 - 4), 1998, p. 159-182.

FENSTER, M.; KRAUS, S. *Coordination Without Communication*: *Experimental Validation of Focal Point Techniques*. Readings in Agents. Michael N. Huhns and Munindar P. Singh (Eds.) chapter 4. San Francisco: Morgan Kaufmann Publishers, 1998, p. 380-386.

FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, Longman Ink., New York, 1999.

FREEMAN, L. C. *Some antecedents of social network analysis*. Connections, v. 19, n. 1, 1996, p. 39-42.

FROZZA, R.; ALVARES, L. O. C. *Criteria for the Analysis of Coordination in Multi-Agent Applications*. In: Coordination Models and Languages - Coordination, York. Lecture Notes in Computer Science 2315. 2002, p. 158-165.

GÁMEZ, J. A.; PUERTA, J. M. *Searching for the best elimination sequence in Bayesian networks by using ant colony optimization*. Pattern Recognition Letters (23), 2002, p. 261-277.

GAMBARDELLA. L. M.; DORIGO, M. *Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem.* Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, 1995, p. 252-260.

GAMBARDELLA, L. M.; TAILLARD, E. D.; DORIGO, M. *Ant Colonies for the QAP*. Technical report, IDSIA, Lugano, Switzerland, 1997a.

GAMBARDELLA, L. M.; DORIGO, M. *HAS-SOP: Hybrid ant system for the sequential ordering problem*. Technical Report IDSIA, IDSIA, Lugano, Switzerland, 1997b, p. 11-97.

GAMBARDELLA, L. M.; TAILLARD, E. D.; AGAZZI, G. *MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows.* In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, London, UK, 1999a, p. 63-76.

GAMBARDELLA, L. M.; TAILLARD, E. D.; DORIGO, M. *Ant Colonies for the Quadratic Assignment Problems*. Journal of Operational Research Society, v. 50, 1999b, p. 167-176.

GASTON, M. E.; DesJARDINS, M. *Social Network Structures and their Impact on Multi-agent System Dynamics*. In Proceedings of the 18th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS-05), Clearwater Beach, FL, 2005, p. 32-37.

GMYTRASIEWICZ, P. J.; DURFEE, E. H. *A rigorous, operational formalization of recursive modeling*. In V. Lesser (Ed.), Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), Cambridge, MA: MIT Press, 1995, p. 125-132.

GOSS, S.; ARON, S.; DENEUBOURG, J. L.; PASTEELS, J. M. *Self-organized shortcuts in the Argentine ant.* Naturwissenschaften, v. 76, 1989, p. 579-581.

GRANOVETTER, M. S. *The strength of weak ties*. American Journal of Sociology, v. 78, n. 6, 1973, p. 1360-1380.

GROSSER, K. *Human networks in organizational information processing.* Annual Review of Information Science and Technology, Charlotte, v. 26, 1991, p. 349-402.

GROSZ, B. J.; HUNSBERGER, L.; KRAUS, S. *Planning and Acting Together*, AI Magazine Volume 20 N. 4, 1999, (AAAI).

GROSZ, B. J.; KRAUS, S. *Collaborative Plans for Complex Group Action*. Artificial Intelligence 86(2): 269-357, 1996.

GUNTSCH, M.; MIDDENDORF, M. *Pheromone Modification Strategies for Ant Algorithms Applied to Dynamic TSP*. In Proceedings of the Workshop on Applications of Evolutionary Computing, 2001, p. 213-222.

HADAD, M.; KRAUS, S. *SHAREDPLANS in Electronic Commerce*. In Intelligent Information Agents,ed. M. Klusch, Heidelberg, Germany: Springer-Verlag, 1999, p. 204-231.

HADAD, M.; KRAUS S. *Exchanging and Combining Temporal Information in a Cooperative Environment*. Cooperative Information Agents (CIA 2002), Madrid, Spain. Lecture Notes in Computer Science, v. 2446, 2002, p. 279-286.

HENDLER, J. A. *Intelligent Agents: Where AI Meets Information Technology.* IEEE Expert, v. 11, n. 6, 1996, p. 20-23.

HUHNS, M. N.; STEPHENS, L. M. *Multiagent Systems and Societies of Agents*. In: Weiss, Gerhard (Ed.). Multiagent Systems - A modern Approach. [S.I.]: MIT Press, 1999.

HUISMAN, M.; VAN DUIJN, M. A. J. *StOCNET: Software for the statistical analysis of social networks*. Connections, 25(1), 2003, p. 7-26.

HUISMAN, M.; VAN DUIJN, M. A. J. *Software for statistical analysis of social networks*. Proceedings of the Sixth International Conference on Logic and Methodology. Amsterdam, The Netherlands. 2004.

JENNINGS, N. R. *Coordination Techniques for Distributed Artificial Intelligence.* Foundations of Distributed Artificial Intelligence. O'HARE, G.M.P. and JENNINGS, N. R. (Eds.). 1996, p. 187-210.

JENNINGS, N. R.; BUSSMANN, S. *Agent-Based Control Systems.* IEEE Control Systems Magazine, 2003, p. 61-73.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. *Reinforcement learning: A survey*. Journal of Artificial Intelligence Research, v. 4, 1996, p. 237-285.

KAJI, T. *Approach by Ant Tabu Agents for Traveling Salesman Problem.* In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, v. 5, 2001, p. 3429-3434.

KENNEDY, J. F.; EBERHART, R. C. *Particle swarm optimization*. In: Proceedings of the IEEE international conference on neural networks, vol. 4. Perth, Australia; 1995, p. 1942-1948.

KENNEDY, J. F.; EBERHART, R. C.; SHI, Y. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.

KNOKE, D.; YANG, S. *Social Network Analysis*. 2nd ed., Series: Quantitative Applications in the Social Sciences, Sage Publications, Inc, 2008.

KRACKHARDT, D.; BLYTHE, J.; MCGRATH, C. *KrackPlot 3: An improved network drawing program*. Connections. 17: 1994, p. 53-55.

KRAUS, S. *Strategic Negotiation in Multiagent Environments*. MIT Press, Cambridge, USA, 2001.

LATANÉ, B. *The psychology of social impact* American Psychologist, v. 36, n. 4, 1981, p. 343-356.

LEE, S. G.; JUNG, T. U.; CHUNG, T. C. *An Effective Dynamic Weighted Rule for Ant Colony System Optimization*. In Proceedings of the IEEE Congress on Evolutionary Computation, 2001a, p. 1393-1397.

LEE, S. G.; JUNG, T. U.; CHUNG, T. C. *Improved Ant Agents System by the Dynamic Parameter Decision*. In Proceedings of the IEEE International Conference on Fuzzy Systems, 2001b, p. 666-669.

LEE, Z. J.; LEE, C. Y. *A hybrid search algorithm with heuristics for resource allocation problem*. Information Sciences, v. 173, 2005, p. 155-167.

LESSER, V. R.; ORTIZ, C. L.; TAMBE, M. *Distributed Sensor Networks: a Multiagent Perspective*. Massachusetts, New York: Kluwer Academic Publishers, v. 9, 2003, p. 11-20.

LI, Y.; GONG, S. *Dynamic Ant Colony Optimization for TSP*. International Journal of Advanced Manufacturing Technology, 22(7-8): 2003, p. 528-533.

LIM, A.; LIN, J.; RODRIGUES, B.; XIAO, F. *Ant colony optimization with hill climbing for the bandwidth minimization problem*. Applied Soft Computing, v. 6, Issue 2, 2006, p. 180-188.

LITTMAN, M. L. *Markov games as a framework for multi-agent reinforcement learning*. In: Proceedings of the 11th International Conference on Machine Learning (ICML-94). New Brunswick, NJ: Morgan Kaufmann, 1994, p. 157-163.

LITTMAN, M. L.; KAELBLING, L.P. *Reinforcement Learning: A Survey*. Journal of Intelligence Research 4, 1996, p. 237-285.

LIU, J. S.; SYCARA, K. *Multiagent Coordination in Tightly Coupled Task Scheduling*. In Tokoro, M., editor, Proceedings of the Second International Conference on Multi-Agent Systems, Menlo Park, California. AAAI Press, 2001, p. 181-188.

MAES, P. *Artificial Life Meets Entertainment: Lifelike Autonomous Agents*, Communications of ACM, v. 38, n. 11, 1995, p. 108-114.

MAHADEVAN, S.; CONNELL, J. *Automatic programming of behavior-based robots using reinforcement learning*. Artificial Intelligence, Elsevier Science Publishers Ltd., v. 55, 1992, p. 311-365.

MAILLER, R.; LESSER, V. R. *Solving Distributed Constraint Optimization Problems Using Cooperative Mediation*. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent (AAMAS), EUA. Washington DC: IEEE Transactions, 2004, p. 438-445.

MANIEZZO, V.; CARBONARO, A. *An ANTS heuristic for the frequency assignment problem*. Future Generation Computer Systems, 2000, p. 927-935.

MATARIC, M. J. *Using Communication to Reduce Locality in Distributed Multi-Agent Learning*, Journal of Experimental and Theoretical Artificial Intelligence, special issue on Learning in DAI Systems, Gerhard Weiss, ed., 10(3), 1998, p. 357-369.

MAZZEO, S.; LOISEAU, I. *An Ant Colony Algorithm for the Capacitated Vehicle Routing*. Electronic Notes in Discrete Mathematics, 2004, p. 181-186.

MÉRIDA-CAMPOS, C.; WILLMOTT, S. *Modelling Coalition Formation over Time for Iterative Coalition Games*. Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04), v. 2, 2004, p. 572-579.

MICHEL, R.; MIDDENDORF, M. *An island based ant system with lookahead for the shortest common supersequence problem*. In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, editors, Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature, volume 1498 of LNCS, Springer Verlag, 1998, p. 692-708.

MILGRAM, S. *The small word problem.* Psychology Today, v. 2, 1967, p. 60-67.

MITCHELL, J. C. *The concept and use of social networks*, in Mitchell, J.C. (Ed.), Social Networks in Urban Situations: Analyses of Personal Relationships in Central African Towns, Manchester University Press, Manchester, 1969, p. 1-50.

MITCHELL, T. *Machine learning*. New York: McGraw Hill, 1997.

MODI, P. J.; SHEN, W. *Collaborative Multiagent Learning for Classification Tasks*. In Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press. Montreal - Quebec, Canada, 2001, p. 37-38.

MODI, P. J.; SHEN, W.; TAMBE, M.; YOKOO, M. ADOPT: *Asynchronous Distributed Constraint Optimization with quality guarantees*. Artificial Intelligence 161: 2005, p. 149-180.

MONTEIRO, S. T.; RIBEIRO, C. H. C. *Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel.* Sba Controle & Automação, v. 15, n. 3, 2004, p. 320-338.

MORENO, J. L. *Who shall survive? Foundations of sociometry,group psychotherapy and sociodrama*. Inc. Beacon New York, 3rd edition 1978.

MOULIN, B.; CHAIB-DRAA, B. *An Overview of Distributed Artificial Intelligence.* In: O'hare, Greg; Jennings, Nicholas R. (Eds.). Foundations of distributed artificial intelligence. [S.I.]: John Wiley and Sons, N.Y, 1996.

NOH, S.; GMYRASIEWICZ, P. J. *Multiagent coordination in antiair defense: A case study*. In M. Boman and W. V. de Velde, editors, Multi-Agent Rationality - MAAMAW'97 Workshop, Lecture Notes in Artificial Intelligence, v. 1237, Springer, New York, 1997, p. 4-16.

NWANA, H. S.; LEE, L.; JENNINGS, N. R. *Coordination in Software Agent Systems*. BT Technology Journal, v. 14 (4), 1996, p. 79-88.

OGDEN, B.; DAUTENHAHN, K. *Embedding robotic agents in the social environment*, Proc. TIMR 2001, Towards Intelligent Mobile Robots. 2001.

OSSOWSKI, S. *Co-ordination in Artificial Agent Societies*, Social Structure and its Implications for Autonomous Problem-Solving Agents, LNCS, v. 1535, 1999.

PANZARASA, P.; JENNINGS, N. R. *The organization of sociality: a manifesto for a new science of multi-agent systems*, Proceedings of the 10th European Workshop on Multi-agent Systems, (MAAMAW01), Annecy, France, 2001.

PAVLOV, I. P. *Conditioned Reflexes*. London: Oxford University Press, England, 1927.

PEARL, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

PENG, J.; WILLIAMS, R. J. *Incremental multi-step Q-Learning.* W. W. Cohen e H. Hirsh (eds.), Proceedings of the Eleventh International Conference on Machine Learning, San Francisco: Morgan Kaufmann, 1996, p. 226-232.

PETCU, A.; FALTINGS, B *A scalable method for multiagent constraint optimization*. In *IJCAI 05*, Edinburgh, Scotland, 2005, p. 266-271.

PORTA, J. M.; CELAYA, E. *Reinforcement Learning for Agents with Many Sensors and Actuators Acting in Categorizable Environments*. Journal of Artificial Intelligence Research, v. 23, 2005, p. 79-122.

RABUSKE, M. A. *Introdução à teoria dos grafos*. Editora UFSC, Florianópolis, Santa Catarina, 1992.

RADCLIFFE-BROWN, A. R. *On Joking Relationships*. Africa, Journal of the International African Institute, vol. 13, no. 3, Jul. 1940, p. 195-210.

RAIFFA, H. *The Art and Science of Negotiation*. Belknap Press; New edition, Paperback, 1985.

REINELT, G. *TSPLIB – A traveling salesman problem library*, ORSA Journal on Computing, 3, 376 - 384, 1991.

RIBEIRO, C. H. C. *A Tutorial on reinforcement learning techniques*. Supervised Learning track tutorials of the 1999 International Joint Conference on Neuronal Networks. Washington: INNS Press. 1999.

RIBEIRO, C. H. C. *Reinforcement learning agents*. Artificial Intelligence Review, v. 17, 2002, p. 223-250.

RIBEIRO, R.; ENEMBRECK, F.; KOERICH, A. L. *A Hybrid Learning Strategy for Discovery of Policies of Action*. International Joint Conference X Ibero-American Artificial Intelligence Conference (IBERAMIA 2006) and XVIII Brazilian Artificial Intelligence Symposium (SBIA 2006), Ribeirão Preto, SP, Brazil. LNCS, v. 4140, 2006a, p. 268-277.

RIBEIRO, R.; ENEMBRECK, F.; KOERICH, A. L. *Uma Nova Metodologia para Avaliação da Performance de Algoritmos Baseados em Aprendizagem por Reforço*. XXXIII SEMISH, Campo Grande, MS, 2006b, p. 433-446.

RIBEIRO, R. Avaliação e Descoberta de Políticas de Ação para Agentes Autônomos Adaptativos. Dissertação de Mestrado, Programa de Pós-Graduação em Informática Aplicada, PPGIA, Pontifícia Universidade Católica do Paraná, Curitiba, 2006c.

RIBEIRO, R.; KOERICH, A. L.; ENEMBRECK F. *Noise Tolerance in Reinforcement Learning Algorithms*, IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07), Silicon Valley, California, USA, 2007a, p. 265-268.

RIBEIRO, R.; KOERICH, A. L.; ENEMBRECK, F. *Reinforcement Learning: Adaptive Agents for Discovery of Policies of Action,* (Revista Ágora (Caçador)), v. 14, 2007b, p. 9-24.

RIBEIRO, R.; BORGES, A. P.; ENEMBRECK, F. *Interaction Models for Multiagent Reinforcement Learning*. International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA08, Vienna, Austria, 2008a, p. 464-469.

RIBEIRO, R.; BORGES, A. P.; ULBRICH, G.; KOERICH, A. L.; SCALABRIN, E. E.; ENEMBRECK, F. *Discovering of Action Policies in Dynamic Environments* (Revista Ágora (Caçador)), v. 15, 2008b, p. 175-185.

RIBEIRO, R; BORGES, A. P.; RONSZCKA, A. F; ÁVILA, B. C; SCALABRIN, E. E, ENEMBRECK, F. *Cooperação Híbrida em Sistemas Multi-Agente* (Revista de Informática Teórica e Aplicada, 2009a), sobre revisão. 2009.

RIBEIRO. R.; BORGES, A. P.; KOERICH, A.; SCALABRIN, E. E.; ENEMBRECK, F. *A Strategy for Converging Dynamic Action Policies*. In: IEEE Symposium Series on Computational Intelligence, 2009, Nashville. Proceedings of IEEE Symposium Series on Computational Intelligence, v. 10, 2009b, p. 136-143.

RIBEIRO, R.; RONSZCKA, A. F.; BORGES, A. P.; ENEMBRECK, F. *Otimização dos Parâmetros de Aprendizagem para a Coordenação dos Agentes em Algoritmos de Enxames.* Simpósio de Computação Aplicada SCA'09, Passo Fundo, 2009c.

RICHARDS, W. D.; SEARY, A. J. *MultiNet*. Version 4.24 for Windows. Burnaby: Simon Fraser University, 2003.

ROBERTS, F. S. *Applied combinatorics*. Englewood Cliffs: Prentice Hall, 1984.

ROSENSCHEIN, J. S.; GENESERETH, M. R. *Deals among Rational Agents*. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85), 91-99. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, 1985.

ROSENSCHEIN, J. S.; ZLOTKIN, G. *Designing conventions for automated negotiation*. AI Magazine, 1994, p. 29-46.

ROUX, O.; FONLUPT, C.; ROBILLIARD, D.; TALBI, E-G. *ANTabu*, Technical Report LIL-98-04, Laboratoire d'Informatique du Littoral, Université du Littoral, Calais, France, 1998.

ROUX, O.; FONLUPT, C.; ROBILLIARD, D.; TALBI, EG. *ANTabu* - Enhanced Version, Technical Report LIL-99-1, Laboratoire d'Informatique du Littoral, Université du Littoral, Calais, France, 1999.

RUBINSTEIN, R. Y. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Inc., New York, USA, 1st edition, 1981.

SANDHOLM, T.; LARSON, K.; ANDERSSON, M.; SHEHORY, O.; TOHME, F. *Worst-case-optimal anytime coalition structure generation*. In Proceedings of AAAI-98, pages 43-56, 1998, Menlo Park, CA. AAAI Press.

SANDHOLM, T, W.; LESSER, V, R. *Coalition Formation among Bounded Rational Agents*. Computer Science Department, University of Massachusetts, Technical Report: UM-CS-1995-071, 1995.

SCHERMERHORN, P.; SCHEUTZ, M. *Social Coordination without Communication in Multi-Agent Territory Exploration Tasks*. In The Proc. of the Fifth Int. Joint Conference on AAMAS-06, Hakodate, Japan, 2006, p. 654-661.

SCHWARTZ, A. *A reinforcement Learning Method for Maximizing Undiscounted Rewards*. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, Massachusetts. Morgan Kaufmann, 1993, p. 298-305.

SHEHORY, O.; KRAUS, S. *Task allocation via coalition formation among autonomous agents*. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Quebec, Canada, 1995, p. 655-661.

SHYU, S. J.; LIN, B. M. T.; HSIAO, T-S. *Ant colony optimization for the cell assignment problem in PCS networks*. Computers & Operations Research, v. 33, n. 6, 2006, p. 1713-1740.

SICHMAN, J. S. *Raciocínio Social e Organizacional em Sistemas Multiagentes: Avanços e Perspectivas.* Tese (Escola Politécnica da Universidade de São Paulo, para obtenção do título de Professor Livre Docente) - USP, São Paulo, 2003.

SIEGEL, S. *Estatística não Paramétrica*. São Paulo: McGraw Hill, 1975.

SIM, K. M.; SUN, W. H. *Multiple Ant-Colony Optimization for Network Routing.* In Proceedings of the First International Symposium on Cyber Worlds, 2002, p. 277-281.

SINGH, S. P.; SUTTON, R. S. *Reinforcement learning with replacing eligibility traces*. Machine Learning, n. 22, 1996, p. 123-158.

SMITH, R. G. *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Transactions on Computers, v. C-29, 1980, p. 1104-1113.

SNIJDERS, T. A. B. *The statistical evaluation of social network dynamics.* In Sobel, M.E., and Becker, M.P. (eds.), Sociological Methodology, London: Basil Blackwell. 2001, p. 361-395.

SOH, L. K.; LUO, J. *Combining Individual and Cooperative Learning for Multiagent Negotiations*. Proceedings of the 2nd Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03), Melbourne, Australia, 2003, p. 1122-1123.

STOKMAN, F. N.; SPRENGER, C. J. A. *GRADAP: Graph definition and analysis package*. Version 2.0. Groningen: iec. ProGAMMA. 1989.

STONE, P.; VELOSO, M. *Towards Collaborative and Adversarial Learning: a Case Study in Robotic Soccer*. International Journal of Human-Computer Studies, v. 48, issue 1. Evolution and learning in multiagente systems. Academic Press, Inc. Duluth, MN, USA, 1996, p. 83-104.

STUTZLE, T.; HOOS, H. *MAX-MIN Ant System and Local Search for The Traveling Salesman Problem*. In Proceedings of the IEEE International Conference on Evolutionary Computation, 1997, p. 309-314.

STUTZLE, T. *Local search algorithms for combinatorial problems - analysis, improvements, and new applications.* PhD thesis, Department of Computer Science, Darmstadt University of Technology, Darmstad, Germany, 1998.

SU, C. T.; CHANG, C-F.; CHIOU, J-P. *Distribution network reconfiguration for loss reduction by ant colony search algorithm*. Electric Power Systems Research, v. 75, 2005, p. 190-199.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. A Bradford book, The MIT Press, London, England, 1998.

SYCARA, K. P. *Resolving Goal Conflicts via Negotiation*. In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88). Menlo Park, Calif.: American Association for Artificial Intelligence, 1988.

SYCARA, K. P. *Persuasive Argumentation in Negotiation*. Theory and Decisions 28:203-242, 1990.

TADEPALLI, P.; OK, D. *A reinforcement learning method for optimizing undiscounted average reward*. Technical Report, Department of Computer Science, Oregon State University, 1994.

TAILLARD, E. D.; GAMBARDELLA, L. M. *Adaptive Memories for the Quadratic Assignment Problem*. Technical report, IDSIA, Lugano, Switzerland, 1997.

TAILLARD, E. D. *FANT: Fast Ant System*. Technical Report IDSIA-46-98, IDSIA, Lugano, Switzerland, 1998.

TAMBE, M. *Towards Flexible Teamwork*. Journal of Artificial Intelligence Research, v. 7, 1997, p. 83-124.

TESAURO, G. *Temporal difference learning and TD-Gammon*, Communications of the ACM, v. 38 (3), 1995, p. 58-68.

TSAI, C. F.; TSAI, C. W.; WU, H. C.; YANG, T. *ACODF: A novel data clustering approach for data mining in large databases*. The Journal of Systems and Software, 2004, p. 133-145.

VIDAL, J. M. *The effects of cooperation on multiagent search in task-oriented domains*. Journal of Experimental and Theoretical Artificial Intelligence, 16(1):5-18, 2004.

WASSERMAN, S.; FAUST, K. *Social Network Analysis: methods and applications*. Cambridge: Cambridge University Press, 1994.

WATKINS, C. J. C. H.; DAYAN, P. *Q-Learning, Machine Learning*, v.8 (3), 1992, p. 279-292.

WATTS, D. J.; STROGATZ, S. H. *Collective dynamics of small-world networks*, Nature, v. 393, n. 6684, 1998, p. 440-442.

WATTS, D. *SMALL Worlds - The Dynamics of Networks between Order and Randomness*, New Jersey: Princeton University Press, 1999.

WATTS, D. *Six Degrees*. The Science of a Connected Age. New York: W. W. Norton & Company, 2003.

WEISS, G.; SEN, S. *Adaptation and Learning in Multiagent Systems*, Lecture Notes in Artificial Intelligence. Berlin, Germany: Springer-Verlag, v. 1042, 1996, p. 1-21.

WEST, D. B. *Introduction to graph theory*. 2nd. ed. Upper Saddle River: John Wiley, 2001.

WILSON, R. J. *Introduction to graph theory*. 4th ed. Harlow: Prentice Hall, 1996.

WOOLDRIDGE, M.; JENNINGS, N. R. *Intelligent Agents: Theory and Practice*. In Knowledge Engineering Review 10(2), pp. 115-152, 1995.

WOOLDRIDGE, M. *Intelligent agents*. In G. Weiss (ed.), Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 1999.

WOOLDRIDGE, M. J. *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.