

ALLAN RODRIGO LEITE

OTIMIZAÇÃO DISTRIBUÍDA DE  
RESTRICÇÃO BASEADA EM TEORIAS  
DE REDES COMPLEXAS E  
SINCRONIZAÇÃO FÍSICA

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

Curitiba  
2017

ALLAN RODRIGO LEITE

OTIMIZAÇÃO DISTRIBUÍDA DE  
RESTRICÇÃO BASEADA EM  
TEÓRIAS DE REDES  
COMPLEXAS E SINCRONIZAÇÃO  
FÍSICA

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

Área de Concentração: Ciência da Computação

Orientador: Fabrício Enembreck

Curitiba  
2017

Leite, Allan Rodrigo

OTIMIZAÇÃO DISTRIBUÍDA DE RESTRIÇÃO BASEADA EM TEORIAS DE REDES COMPLEXAS E SINCRONIZAÇÃO FÍSICA. Curitiba, 2017.

Tese - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Sistemas Multiagente 2. Otimização Distribuída de Restrição 3. Sincronização em Redes de Osciladores Acoplados I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e Tecnologia. Programa de Pós-Graduação em Informática II - t



Dedico este trabalho à minha esposa e minha família, minha fonte de inspiração e de determinação.

# Agradecimentos

Agradeço ao meu pai, minha mãe e meu irmão pelo amor incondicional, carinho e dedicação. Sou afortunado por pertencer a esta família. Ao meu amor, Ingrid, pela paciência e incentivo para conclusão desta etapa em minha vida. Sem seu apoio, este sonho não seria concretizado. Ao meu orientador, professor Dr. Fabrício Enembreck, pela paciência, excepcional dedicação e por acreditar em meu potencial. Um pesquisador ímpar, seu conhecimento e suas ideias foram fundamentais para o desenvolvimento deste trabalho. À CAPES, pelo subsídio financeiro para realização deste trabalho.

# Sumário

<b>Agradecimentos</b>	ii
<b>Sumário</b>	iii
<b>Lista de Figuras</b>	viii
<b>Lista de Tabelas</b>	x
<b>Lista de Símbolos</b>	xii
<b>Lista de Abreviações</b>	xiv
<b>Resumo</b>	xvi
<b>Abstract</b>	xvii

## Capítulo 1

<b>Introdução</b>	1
1.1 Problema . . . . .	2
1.2 Motivação . . . . .	3
1.3 Hipótese . . . . .	4
1.4 Objetivo Geral . . . . .	5
1.5 Objetivos Específicos . . . . .	5
1.6 Organização do Trabalho . . . . .	5

## Capítulo 2

<b>Fundamentos sobre Problemas de Otimização Distribuída de Restrição</b>	7
2.1 Coordenação Descentralizada . . . . .	7
2.2 Princípios da Otimização de Restrição . . . . .	8
2.2.1 Definição de um DCOP . . . . .	9
2.2.2 Programação por Restrição . . . . .	10
2.2.3 Problema de Satisfação de Restrição . . . . .	10
2.2.4 Problema de Satisfação Distribuída de Restrição . . . . .	11
2.2.5 Problema de Otimização de Restrição . . . . .	11

2.2.6	Problema de Otimização Distribuída de Restrição . . . . .	12
2.3	Representação de um DCOP . . . . .	13
2.3.1	Grafo de restrições . . . . .	13
2.3.2	Grafo fator . . . . .	14
2.3.3	Pseudoárvores . . . . .	15
2.4	Variações de DCOP . . . . .	16
2.4.1	DCOP Quantificado . . . . .	16
2.4.2	DCOP Assimétrico . . . . .	17
2.4.3	DCOP Multi-Objetivo . . . . .	18
2.4.4	DCOP Dinâmico . . . . .	19
2.4.5	DCOP Probabilístico . . . . .	20
2.5	Algoritmos para DCOP . . . . .	21
2.5.1	Algoritmos Baseados em Busca, Inferência e Amostragem . . . . .	23
2.5.2	Algoritmos Completos e Incompletos . . . . .	23
2.5.3	Algoritmos Parcialmente Centralizados e Distribuídos . . . . .	24
2.5.4	Algoritmos Síncronos e Assíncronos . . . . .	25
2.6	Aplicações de DCOP . . . . .	25
2.6.1	Coloração de grafos . . . . .	26
2.6.2	Redes de Sensores, Wireless e Smart-Grids . . . . .	26
2.6.3	Agendamento de Eventos e Alocação de Recursos . . . . .	27
2.6.4	Sincronização, Planejamento e Coordenação de Ações . . . . .	28
2.7	Revisão Crítica . . . . .	28
2.7.1	Evolução dos Algoritmos . . . . .	29
2.7.1.1	Algoritmos Completos . . . . .	29
2.7.1.2	Algoritmos Incompletos . . . . .	31
2.7.2	Comparação dos Algoritmos . . . . .	34
2.7.3	Aplicabilidade dos Algoritmos . . . . .	36
2.8	Considerações Finais . . . . .	37

### Capítulo 3

<b>Fundamentos sobre Sincronização em Rede de Osciladores Acoplados</b>	<b>38</b>	
3.1	Sistemas Dinâmicos . . . . .	38
3.2	Modelos Matemáticos de Sincronização . . . . .	39
3.2.1	Modelo de Winfree . . . . .	39
3.2.2	Modelo de Kuramoto . . . . .	40
3.2.3	Parâmetros de Ordem de Kuramoto . . . . .	42

3.2.4	Tipos de Sincronização . . . . .	43
3.2.5	Condição de Sincronização . . . . .	44
3.3	Teoria de Redes Complexas . . . . .	45
3.3.1	Definição de Redes Complexas . . . . .	45
3.3.2	Redes Regulares . . . . .	46
3.3.3	Redes Aleatórias . . . . .	47
3.3.4	Redes de Mundos Pequenos . . . . .	47
3.3.5	Redes Livre de Escala . . . . .	48
3.4	Sincronização em Redes Complexas . . . . .	49
3.4.1	Representação de Redes Complexas . . . . .	49
3.4.2	Redes Complexas Adaptativas . . . . .	51
3.4.3	Estabilidade em Redes Complexas Sincronizadas . . . . .	52
3.4.4	Expoente de Lyapunov . . . . .	53
3.5	Aplicações da Sincronização . . . . .	54
3.5.1	Sistemas Distribuídos . . . . .	54
3.5.2	Controle de Densidades de Carga e Smart-Grids . . . . .	54
3.5.3	Comunicação em Redes Wireless . . . . .	56
3.5.4	Protocolos de Consenso . . . . .	57
3.5.5	Movimento e Controle Coletivo . . . . .	58
3.6	Considerações Finais . . . . .	59

## Capítulo 4

<b>Otimização Distribuída de Restrição Baseada em Sincronização Física</b>	60
4.1 Correlação entre DCOP e Redes de Osciladores . . . . .	60
4.1.1 Variáveis como Osciladores . . . . .	61
4.1.2 Acoplamentos e Funções de Custo . . . . .	62
4.1.3 Processo de Sincronização . . . . .	63
4.2 COOPT: Coupled Oscillators OPTimization . . . . .	66
4.2.1 Representação do Problema . . . . .	66
4.2.2 Modelo de Comunicação . . . . .	67
4.2.3 Propriedade <i>Anytime</i> . . . . .	68
4.2.4 Descrição do Algoritmo . . . . .	69
4.2.5 Exemplo de Execução . . . . .	72
4.2.6 Coesão das Fases . . . . .	76
4.2.7 Análise de Complexidade . . . . .	76
4.2.7.1 Comunicação . . . . .	77

4.2.7.2	Processamento Local . . . . .	77
4.3	Implementação do COOPT . . . . .	78
4.3.1	Plataforma de Simulação FRODO . . . . .	78
4.3.2	Representação do Problema . . . . .	79
4.3.3	Representação de um Agente . . . . .	81
4.3.4	Arquitetura da Implementação . . . . .	83
4.4	Considerações Finais . . . . .	85
<b>Capítulo 5</b>		
<b>Avaliação Experimental</b> . . . . . 86		
5.1	Metodologia de Avaliação . . . . .	86
5.1.1	Protocolo de Avaliação . . . . .	87
5.1.2	Categoria de Problemas . . . . .	88
5.1.3	Topologias Avaliadas . . . . .	88
5.1.4	Geração dos Problemas . . . . .	89
5.2	Métricas Utilizadas . . . . .	90
5.2.1	Qualidade das Soluções . . . . .	90
5.2.2	Ciclos de Mensagens . . . . .	91
5.2.3	NCCC . . . . .	91
5.2.4	Tamanho das Mensagens e Informações Compartilhadas . . . . .	92
5.3	Parâmetros dos Algoritmos . . . . .	92
5.3.1	Algoritmos DSA e DUCT . . . . .	93
5.3.2	Algoritmo COOPT . . . . .	93
5.4	Resultados Obtidos . . . . .	95
5.4.1	Qualidade das Soluções . . . . .	95
5.4.2	Ciclos de Mensagens . . . . .	104
5.4.3	NCCC . . . . .	104
5.4.4	Tamanho das Mensagens e Informações Compartilhadas . . . . .	105
5.5	Análise de Convergência . . . . .	106
5.6	Análise do <i>Tightness</i> . . . . .	108
5.7	Considerações Finais . . . . .	109
<b>Capítulo 6</b>		
<b>Conclusão</b> . . . . . 111		
6.1	Trabalhos Futuros . . . . .	113

Capítulo 7	
Apêndice	115
Referências Bibliográficas	123

# Lista de Figuras

2.1	Exemplo de um DisCSP representado por meio de um grafo de restrições. Adaptada de (YOKOO et al., 1998). . . . .	11
2.2	Relação entre problemas de satisfação de restrição e problemas de otimização de restrição. . . . .	12
2.3	Representação de um DCOP por meio de grafos. . . . .	14
2.4	Categorias dos algoritmos para DCOP. . . . .	22
2.5	Exemplo de uma rede de sensores. . . . .	26
3.1	Dinâmica de sincronização em uma rede de osciladores globalmente acoplados. Gerado pelo software Mathematica (WOLFRAM, 2003). . . . .	41
3.2	Exemplo de redes complexas com diferentes topologias. Gerado pelo software de computação estatística R (R Development Core Team, 2008). . . . .	46
3.3	Exemplo de uma rede de osciladores globalmente acoplados com diferentes pesos em cada aresta. Adaptada de (DÖRFLER; M.; F., 2013). . . . .	50
3.4	Processo iterativo de evolução em redes complexas adaptativas. Adaptada de (GROSS; BLASIUS, 2008). . . . .	52
3.5	Sincronização do fluxo de densidade de carga em uma rede de osciladores. . . . .	55
3.6	Dinâmica de sincronização temporal em uma rede de osciladores acoplados. . . . .	57
4.1	Exemplo de DCOP e do modelo de comunicação do COOPT. . . . .	67
4.2	Execução do COOPT a partir do problema retratado pela Figura 4.1. . . . .	72
4.3	Fluxo de mensagens compartilhadas entre os agentes a partir da sequência de execução do COOPT retratada pela Figura 4.2. . . . .	75
4.4	Demonstração da tendência de sincronização do COOPT sobre o problema retratado pela Figura 4.2. . . . .	76
4.5	Principais camadas da arquitetura modular do FRODO. Adaptada de (LÉAUTÉ; OTTENS; SZYMANEK, 2009). . . . .	78

4.6	Interface gráfica da plataforma FRODO. . . . .	79
4.7	Exemplo de um DCOP representado em formato XML no FRODO . . . . .	80
4.8	Estrutura de um agente representado em formato XML no FRODO . . . . .	82
4.9	Diagrama de classe da implementação do COOPT no framework FRODO. . . . .	84
5.1	Custo médio das soluções para cada valor avaliado do parâmetro $K$ , de acordo com o número de variáveis e restrições envolvidas nos problemas. . . . .	94
5.2	Comparativo da convergência dos algoritmos de busca local nos experimentos envolvendo 10 variáveis e 90% de densidade e <i>tightness</i> . . . . .	106
5.3	Comparativo da convergência dos algoritmos de busca local nos experimentos envolvendo 200 variáveis e 90% de densidade e <i>tightness</i> . . . . .	107
5.4	Análise sobre impacto do <i>tightness</i> sobre o custo das soluções nos experimentos envolvendo 10 variáveis. . . . .	108
5.5	Análise sobre impacto do <i>tightness</i> sobre o custo das soluções nos experimentos envolvendo 200 variáveis. . . . .	109

# Lista de Tabelas

2.1	Avaliação experimental dos algoritmos para DCOP. . . . .	35
5.1	Parâmetros utilizados para geração aleatória dos problemas. . . . .	90
5.2	Resultados do teste de Friedman para análise da relevância do parâmetro $K$ sobre a qualidade das soluções encontradas. . . . .	94
5.3	Resultados dos experimentos envolvendo problemas com 10 variáveis em redes regulares (CD: 2.38) . . . . .	96
5.4	Resultados dos experimentos envolvendo problemas com 10 variáveis em redes aleatórias (CD: 2.38) . . . . .	97
5.5	Resultados dos experimentos envolvendo problemas com 10 variáveis em redes de mundos pequenos (CD: 2.38) . . . . .	98
5.6	Resultados dos experimentos envolvendo problemas com 10 variáveis em redes livres de escala (CD: 2.38) . . . . .	99
5.7	Resultados dos experimentos envolvendo problemas com 200 variáveis em redes regulares (CD: 1.48) . . . . .	100
5.8	Resultados dos experimentos envolvendo problemas com 200 variáveis em redes aleatórias (CD: 1.48) . . . . .	101
5.9	Resultados dos experimentos envolvendo problemas com 200 variáveis em redes de mundos pequenos (CD: 1.48) . . . . .	102
5.10	Resultados dos experimentos envolvendo problemas com 200 variáveis em redes livres de escala (CD: 1.48) . . . . .	103
7.1	Resultados dos experimentos envolvendo problemas com 50 variáveis em redes regulares (CD: 1.93) . . . . .	115
7.2	Resultados dos experimentos envolvendo problemas com 50 variáveis em redes aleatórias (CD: 1.93) . . . . .	116

7.3	Resultados dos experimentos envolvendo problemas com 50 variáveis em redes de mundos pequenos (CD: 1.93) . . . . .	117
7.4	Resultados dos experimentos envolvendo problemas com 50 variáveis em redes livres de escala (CD: 1.93) . . . . .	118
7.5	Resultados dos experimentos envolvendo problemas com 100 variáveis em redes regulares (CD: 1.48) . . . . .	119
7.6	Resultados dos experimentos envolvendo problemas com 100 variáveis em redes aleatórias (CD: 1.48) . . . . .	120
7.7	Resultados dos experimentos envolvendo problemas com 100 variáveis em redes de mundos pequenos (CD: 1.48) . . . . .	121
7.8	Resultados dos experimentos envolvendo problemas com 100 variáveis em redes livres de escala (CD: 1.48) . . . . .	122

## Lista de Símbolos

$\delta(d)$	Custo local quando um agente atribui o valor $d$
$\delta^*$	Custo global de uma solução completa
$\gamma_i$	Ângulo da fase do oscilador $i$
$\Omega$	Frequência média de oscilação do sistema
$\omega_i$	Frequência do oscilador $i$
$\omega_{max}$	Maior frequência natural do sistema
$\omega_{min}$	Menor frequência natural do sistema
$\psi$	Fase média de oscilação do sistema
$\Theta$	Diferença das fases de oscilação do sistema
$\theta_i$	Fase do oscilador $i$
$A$	Solução completa
$a_i$	Agente $i$
$a_{ij}$	Elemento da matriz de adjacência
$d$	Possível valor para uma dada variável
$D_i$	Domínio da variável $x_i$
$d_i$	Valor corrente da variável $i$
$f_{ij}$	Função de custo ou restrição
$K$	Força de acoplamento do sistema

$K_c$	Acoplamento crítico necessário para a emergência de sincronização do sistema
$r$	Parâmetro complexo de coerência do sistema
$x_i$	Variável $i$

# Lista de Abreviações

- ACLS** Asymmetric Coordinated Local Search
- ADCOP** Asymmetric Distributed Constraint Optimization Problem
- ADOPT** Asynchronous Distributed Optimization
- ADOPT-ng** Asynchronous Distributed Optimization - NoGood
- AFB** Asynchronous Forward-Bounding
- ConcFB** Concurrent Forward Bounding
- COOPT** Coupled Oscillators OPTimization
- COP** Constraint Optimization Problem
- CP** Constraint Programming
- CSP** Constraint Satisfaction Problem
- DaCSA** Divide and Coordinate Subgradient Algorithm
- DALO** Distributed Asynchronous Local Optimization
- DBA** Distributed Breakout Algorithm
- DCOP** Distributed Constraint Optimization Problem
- DisCSP** Distributed Constraint Satisfaction Problem
- DJAO** Distributed Junction And/Or Graph
- DPOP** Dynamic Programming Optimization Protocol
- DSA** Distributed Stochastic Algorithm

**DUCT** Distributed Upper Confidence Tree

**D-Gibbs** Distributed Gibbs

**GCA-MGM** Guaranteed Convergence Asymmetric Maximum Gain Message

**IAD** Inteligência Artificial Distribuída

**LS-RAND** Local Search - Random

**MCA** Multiply-Constrained ADOPT

**MGM** Maximum Gain Message

**MSF** Master Stability Function

**NCBB** No-Commitment Branchand-Bound

**NE** Equilíbrio de Nash

**OptAPO** Optimal Asynchronous Partial Overlay

**QCSP** Quantified Constraint Satisfaction Problem

**QDCOP** Quantified Distributed Constraint Optimization Problem

**SCA** Stochastic Coordination Algorithm

**SynchBB** Synchronous Branch-and-Bound

**TDMA** Time Division Multiple Access

# Resumo

Soluções para Problema de Otimização Distribuída de Restrição (DCOP) têm emergido como uma elegante técnica de coordenação em sistemas multiagente, capaz de modelar naturalmente um grande número de problemas de otimização por meio de um conjunto de restrições. Entretanto, devido ao fato do DCOP pertencer à classe de problemas NP-completos, algumas categorias de problemas reais não são aplicáveis a este formalismo em função da complexidade do espaço de soluções do problema e da falta de escalabilidade dos algoritmos para DCOP encontrados na literatura.

Neste sentido, a investigação de novas estratégias baseadas em fundamentações teóricas alternativas é essencial para impulsionar a evolução desta linha de pesquisa, sobretudo ao considerar ambientes complexos e em larga escala. Por sua vez, a sincronização é um fenômeno físico natural que descreve a emergência do comportamento coletivo em sistemas dinâmicos complexos. Modelos de sincronização física em redes de osciladores acoplados têm sido intensamente investigados nas últimas décadas em virtude da capacidade de convergência, escalabilidade e aplicabilidade em inúmeros cenários reais. Estes modelos são flexíveis e suportam uma grande variedade de padrões de sincronização, incluindo técnicas de otimização e dinâmicas complexas de interação entre os elementos.

Este trabalho tem por objetivo introduzir uma nova abordagem para resolução de DCOP baseada em modelos de sincronização física em redes complexas. A principal contribuição deste trabalho consiste em conciliar as características de escalabilidade e adaptabilidade dos modelos de sincronização física nesta nova abordagem para DCOP. Os resultados experimentais obtidos demonstraram que a abordagem proposta é capaz de endereçar problemas reais em larga escala de maneira mais eficiente que os principais métodos para resolução de DCOP existentes até então.

**Palavras-chave:** Sistemas Multiagente; Problema de Otimização Distribuída de Restrição; Sincronização em Redes de Osciladores Acoplados; Modelo de Kuramoto.

# Abstract

The Distributed Constraint Optimization Problem (DCOP) has emerged as one of the most promising coordination techniques in multiagent systems, owing to its ability for modeling a large class of optimization problems naturally from a set of constraints. However, because DCOP is known to be NP-Hard, some real applications are unsuitable to this formalism due to the space-state complexity and the lack of scalability of the existing DCOP algorithms.

Therefore, the investigation of new strategies using alternatives theoretical foundation is essential to propel the evolution of this important research area, especially when considering complex and large-scale environments. In this sense, synchronization is a natural physical phenomenon that describes the spontaneous emergence of collective behavior in complex dynamic systems. Synchronization in coupled oscillators networks have been intensely studied in recent decades because of the robustness and applicability of these models in large-scale and real-world problems. Moreover, these models are flexible enough to be adapted to different contexts and support a large variety of synchronization patterns, including optimization techniques and complex interaction dynamics.

This study aims to introduce a new approach for solving DCOP using synchronization models in complex networks. Therefore, the main contribution of this study is to capture the scalability and convergence features found in physical synchronization models and reproduce them into this new approach. Experimental results showed that the proposed approach is able to handle real large-scale problems and outperforms other state-of-the-art DCOP techniques.

**Keywords:** Multi-agent Systems; Distributed Constraint Optimization Problem; Synchronization in Coupled Oscillators Network; Kuramoto Model.

# Capítulo 1

## Introdução

Sistemas multiagente representam um importante paradigma de análise, projeto e implementação de sistemas complexos (WOOLDRIDGE; JENNINGS, 1995). Agentes de software e sistemas baseados em agentes têm sido amplamente utilizados em uma grande variedade de aplicações reais, sobretudo em ambientes complexos envolvendo aspectos como larga escala, tempo real ou missão crítica. Cenários reais como manufatura e controle de processos, gestão da informação, controle de tráfego aéreo, comércio eletrônico, telecomunicações e sistemas de transporte são exemplos de aplicações candidatas para este paradigma (JENNINGS; BUSSMANN, 2003).

Uma característica tipicamente presente em um sistema multiagente é o controle descentralizado, onde os agentes devem coordenar suas ações de maneira distribuída a fim de alcançarem seus objetivos (JENNINGS; SYCARA; WOOLDRIDGE, 1998). Esta característica tem como objetivo assegurar a escalabilidade, reduzir pontos de falhas e otimizar o uso dos recursos computacionais do sistema. No entanto, a coordenação descentralizada constitui um dos maiores desafios em um sistema multiagente, devido à natureza autônoma dos agentes e seus interesses individuais possivelmente conflitantes.

Soluções para Problemas de Otimização Distribuída de Restrição (DCOP) têm emergido como uma das principais técnicas para lidar com os desafios da coordenação descentralizada em sistemas multiagente, em virtude da capacidade de modelar naturalmente diversas categorias de problemas do mundo real (PETCU, 2007). A principal motivação para utilização deste formalismo como modelo de coordenação em sistemas multiagente está relacionada com o fato dos métodos para resolução de DCOP serem distribuídos e robustos (ROSSI; BEEK; WALSH, 2006).

Por definição, um DCOP consiste de um conjunto de variáveis e restrições distribuídas entre agentes, onde cada variável possui um domínio discreto e finito. Os valores de um domínio representam os possíveis ações ou estados que um agente pode assumir.

Por sua vez, as restrições denotam funções de custo sobre as possíveis atribuições de um subconjunto de variáveis. O objetivo de um DCOP consiste em encontrar uma combinação de ações entre todos os agentes que minimizem o custo global do problema (LESSER; ORTIZ; TAMBE, 2003). Todavia, devido ao fato do DCOP pertencer à classe de problemas NP-completos, o uso de estratégias eficientes é indispensável para reduzir a exploração de estados durante o processo de resolução do problema (MODI et al., 2005), em especial em ambientes complexos e em larga escala.

## 1.1 Problema

A linha de pesquisa sobre DCOP tem atraído a atenção de diversos pesquisadores nos últimos anos. De maneira geral, o foco recente das pesquisas está centrado no desenvolvimento de novos algoritmos para DCOP utilizando diferentes estratégias de resolução. Não obstante, cada estratégia de resolução possui limitações que normalmente estão associadas às características do domínio do problema. Em virtude destas limitações conhecidas, a escolha adequada do algoritmo para DCOP é fundamental para obter um desempenho satisfatório em cenários reais.

Uma questão em aberto sobre esta linha de pesquisa está associada à escalabilidade dos algoritmos atuais para DCOP. Alguns cenários reais de sistemas multiagente em larga escala, tais como redes de sensores, podem envolver milhares de variáveis ou restrições (LESSER; ORTIZ; TAMBE, 2003; SCERRI; VINCENT; MAILLER, 2010). No entanto, os algoritmos atuais para DCOP não são favoráveis para problemas desta magnitude em função do esforço computacional requerido pelo processo de resolução ou da baixa qualidade das soluções encontradas (LEITE; ENEMBRECK; BARTHÈS, 2014).

Outra crítica relacionada aos algoritmos atuais para DCOP encontra-se na falta de diversidade teórica das estratégias de resolução. Em termos gerais, os algoritmos implementam estratégias de resolução bastante conhecidas e exploradas na literatura, resultando em um desempenho e comportamentos muito similares, sem uma contribuição efetiva para esta linha de pesquisa. Neste sentido, a investigação de novas estratégias de resolução baseadas em fundamentações teóricas alternativas é essencial para impulsionar a evolução das pesquisas sobre DCOP.

## 1.2 Motivação

Um tema pouco explorado em pesquisas sobre DCOP refere-se ao estudo analítico das particularidades de um determinado ambiente por meio das variáveis, dos domínios e das restrições definidas sobre o problema. Esta análise pode ser estática, abordando questões estruturais como a topologia do problema, ou dinâmica, considerando aspectos comportamentais sobre as relações entre as variáveis e suas implicações no problema. A partir destes estudos é possível, por exemplo, determinar a tendência coletiva do sistema ou identificar a influência e relevância de cada agente sobre o problema.

A teoria de redes complexas (WASSERMAN; FAUST, 1994) e os modelos de sincronização física em redes de osciladores acoplados (KURAMOTO, 1984) possibilitam descrever a natureza das ligações entre os elementos de uma determinada rede, bem como compreender o comportamento coletivo que emerge a partir da interação dos elementos, haja vista que tal comportamento coletivo não pode ser explicado exclusivamente por meio da análise individual de cada elemento. Portanto, a compreensão da dinâmica envolvida na interação dos elementos em uma rede é fundamental para a definição de modelos de coordenação eficazes para sistemas multiagente em larga escala.

Diversos sistemas dinâmicos reais apresentam a tendência de atingir um consenso rítmico através do comportamento coletivo. A sincronização é um fenômeno físico natural que descreve a emergência do comportamento coletivo em sistemas dinâmicos complexos. Em outras palavras, a sincronização física é a manifestação coletiva da habilidade de auto-organização observada em sistemas dinâmicos complexos (ACEBRÓN et al., 2005). Este fenômeno constitui uma das principais linhas de pesquisa sobre sistemas dinâmicos, cujo objetivo consiste em compreender e determinar a tendência de um dado sistema em operar sincronamente ao longo do tempo.

Uma das tentativas mais bem-sucedidas em descrever o fenômeno da sincronização física é conhecida como modelo de Kuramoto (KURAMOTO, 1975). Este modelo matemático possibilita reproduzir o comportamento coletivo emergente em uma rede de osciladores acoplados. Um oscilador é um dispositivo que produz sinais repetitivos de maneira autônoma. Quando o estado interno de um oscilador atinge um certo limite, é emitido um pulso para todos os demais osciladores acoplados. O efeito deste pulso emitido visa alterar o estado atual dos demais osciladores acoplados a fim de aumentar o consenso rítmico entre eles.

Como consequência, o comportamento coletivo descrito pelo modelo de Kuramoto conduz os osciladores de uma rede à sincronização espontânea de seus estados ao longo do tempo. Uma das características mais notáveis do modelo de Kuramoto é, de fato, sua

capacidade de descrever o processo de convergência de uma rede até alcançar o estado síncrono entre os osciladores, seja em redes complexas, em larga escala ou envolvendo diferentes padrões de conexão entre os elementos.

### 1.3 Hipótese

Estudos relacionados à dinâmica coletiva emergente e processos de sincronização física em redes complexas podem contribuir para a evolução das pesquisas sobre DCOP. Nesta perspectiva, o processo de sincronização física em redes de osciladores acoplados, mais especificamente, o modelo de Kuramoto, pode surgir como uma fundamentação teórica alternativa para novas estratégias de resolução de DCOP e contribuir com algumas questões em aberto deste campo de pesquisa, sobretudo em relação à aplicabilidade deste formalismo em categorias de problemas complexos reais e em larga escala.

A atividade rítmica de um oscilador pode ser descrita como uma variável física que evolui ao longo do tempo. Uma das maneiras de representar a emergência do comportamento coletivo do sistema consiste em supor que, quando esta variável atinge um determinado limiar, será emitido um pulso para as variáveis vizinhas ajustarem coletivamente seus ritmos. Em seguida, a variável é reiniciada e este ciclo se repete. Esta analogia da sincronização física apresenta elementos e características muito similares a um DCOP, visto que em ambos os casos, os objetivos estão relacionados à coerência global, ou consenso, entre grupos de indivíduos interagindo entre si.

Portanto, a investigação de modelos de sincronização física em redes complexas pode surgir como uma estratégia alternativa para as pesquisas sobre DCOP, por meio da compreensão da dinâmica envolvida na coordenação entre os elementos da rede. Os modelos de sincronização física em redes de osciladores acoplados, em especial o modelo de Kuramoto, são suficientemente robustos para representar uma grande variedade de padrões de sincronização e suficientemente flexíveis para se adaptar a diferentes contextos de aplicação (ARENAS et al., 2008).

Neste sentido, o modelo de Kuramoto é a fonte de inspiração deste trabalho para o desenvolvimento de um novo método para resolução de DCOP baseado em sincronização física. Mais especificamente, a hipótese deste trabalho reside na possibilidade de capturar a essência do modelo de Kuramoto e aplicá-la em um novo método para resolução de DCOP. No entanto, esta correlação deve preservar as propriedades do formalismo DCOP, como por exemplo, a definição de variáveis e restrições, visão parcial dos agentes sobre o problema, privacidade de informação, dentre outras.

## 1.4 Objetivo Geral

Este trabalho visa o estudo e desenvolvimento de um novo método para resolução de DCOP baseado em modelos de sincronização física em redes de osciladores acoplados. Além disto, este novo método proposto deve ser capaz de endereçar problemas complexos reais em larga escala. Assim, espera-se que o potencial de escala existente em modelos de sincronização física também seja aplicável ao contexto de um DCOP. Além de propor um novo método para resolução de DCOP, outra contribuição deste trabalho consiste em estabelecer uma nova abordagem teórica para DCOP a fim de estimular novas perspectivas de pesquisa sobre o tema.

## 1.5 Objetivos Específicos

A fim de alcançar o objetivo geral proposto para este trabalho, destacam-se os seguintes objetivos específicos:

- Compreender os princípios do processo de sincronização física e a emergência do comportamento coletivo em redes de osciladores acoplados;
- Elaborar um novo método para resolução de DCOP utilizando como fundamentação teórica os conceitos da sincronização física;
- Implementar o método proposto para resolução de DCOP baseado em sincronização física em um ambiente de simulação; e
- Avaliar a eficiência, a escalabilidade e a qualidade das soluções encontradas no novo método proposto, comparando os resultados com os principais algoritmos para DCOP existentes até então na literatura.

## 1.6 Organização do Trabalho

Este trabalho é composto por seis capítulos. O primeiro fornece uma introdução onde são contextualizados o problema, motivações e a hipótese deste projeto. O Capítulo 2 introduz os fundamentos sobre coordenação descentralizada em sistemas multiagente, com ênfase em problemas de otimização distribuída de restrição, aplicações e os principais métodos de resolução para DCOP. O Capítulo 3 apresenta conceitos sobre sistemas dinâmicos não-lineares, modelos matemáticos de sincronização, redes complexas e exemplos reais de modelos de sincronização em sistemas distribuídos. O Capítulo 4 descreve um

método para resolução de DCOP baseado em princípios de modelos de sincronização física. Já o Capítulo 5 apresenta a metodologia de avaliação do método proposto, bem como os resultados obtidos por meio de avaliações experimentais. Finalmente, o Capítulo 6 encerra a discussão com as considerações finais sobre o trabalho, sugestões para trabalhos futuros, direcionamento e perspectivas para novas pesquisas.

## Capítulo 2

# Fundamentos sobre Problemas de Otimização Distribuída de Restrição

Agentes inteligentes é uma área de pesquisa da Inteligência Artificial Distribuída (IAD) intensamente estudada desde a década de 70. O desenvolvimento de aplicações baseadas em agentes tem recebido muita atenção devido à natureza distribuída e autônoma das entidades de software. Por definição, um agente de software possui inteligência sobre um domínio específico e uma visão parcial do problema, permitindo que o agente atue e avalie o impacto de suas ações sobre o ambiente (WOOLDRIDGE; JENNINGS, 1995).

Além disto, um agente deve ser capaz de raciocinar e tomar ações de maneira autônoma. Entretanto, devido à visão parcial sobre o problema e as possíveis dependências existentes entre os subproblemas, os agentes devem cooperar entre si para resolver seus problemas locais. Um dos principais desafios inerente ao desenvolvimento de sistemas multiagente está associado à capacidade dos agentes cooperarem e coordenarem suas ações para alcançar um estado estável sobre o problema.

Este capítulo fornece uma introdução sobre um modelo eficiente para coordenação em sistemas multiagente, denominado Problema de Otimização Distribuída de Restrição, ou *Distributed Constraint Optimization Problem* (DCOP). Neste capítulo também são apresentados os principais algoritmos deste formalismo, exemplos de aplicações reais, bem como uma revisão crítica e perspectivas desta linha de pesquisa.

## 2.1 Coordenação Descentralizada

Sistemas multiagente são tipicamente projetados considerando a ausência de um controle central, isto é, os agentes devem ser capazes de coordenar suas ações de forma distribuída para eficientemente alcançarem seus objetivos individuais (WEISS, 1999). Como benefício, tal característica promove a escalabilidade e reduz eventuais pontos de

falhas em um sistema multiagente, além de otimizar o uso da capacidade computacional dos dispositivos envolvidos (LESSER; ORTIZ; TAMBE, 2003).

Neste cenário, a coordenação descentralizada surge como um dos aspectos chave em sistemas multiagente. Entretanto, a definição do modelo de coordenação a ser utilizado em um sistema multiagente não é uma tarefa trivial, em virtude do comportamento autônomo dos agentes e seus interesses frequentemente distintos (FERBER, 1999). De maneira geral, os desafios da coordenação descentralizada envolvem diversos aspectos como comunicação, controle descentralizado, conhecimento parcial do problema e interesses individuais dos agentes (SCERRI; VINCENT; MAILLER, 2010). Existem diferentes abordagens para lidar com estes desafios, tais como:

- Coordenação reativa (FERBER, 1999; JENNINGS; BUSSMANN, 2003);
- Aprendizagem de máquina (SUTTON; BARTO, 1998; WOOLDRIDGE, 2009);
- Formação de coalizão (SANDHOLM; LESSER, 1995; WEISS, 1999);
- Planejamento (O'HARE; JENNINGS, 1996; SCERRI; VINCENT; MAILLER, 2010);
- Negociação (SHOHAM; LEYTON-BROWN, 2009).

O Problema de Otimização Distribuída de Restrição, ou *Distributed Constraint Optimization Problem* (DCOP), têm emergido como um importante formalismo para coordenação descentralizada em sistemas multiagente devido sua capacidade de modelar naturalmente diversas categorias de problemas do mundo real. As motivações para a utilização deste formalismo como modelo de coordenação em sistemas multiagente referem-se ao fato dos métodos para resolução de DCOP serem essencialmente distribuídos e robustos (ROSSI; BEEK; WALSH, 2006).

## 2.2 Princípios da Otimização de Restrição

Um DCOP é composto por um conjunto de variáveis distribuídas entre agentes, onde cada variável possui um domínio discreto e finito. Por sua vez, cada valor contido no domínio de uma variável representa uma possível ação que o agente associado à variável pode realizar. No entanto, as variáveis normalmente possuem restrições sobre os valores que podem assumir. Estas restrições representam relações de custo ou recompensa entre um subconjunto de atribuições de variáveis (LESSER; ORTIZ; TAMBE, 2003).

O objetivo de um DCOP visa encontrar uma combinação de ações dos agentes que otimize a função objetivo global definida para o problema (MODI et al., 2005). Não

obstante, este formalismo pertence a classe de complexidade NP-completo, visto que se trata de problemas de natureza combinatória, tais como satisfatibilidade ou otimização (GAREY; JOHNSON, 1979). Neste sentido, estratégias eficientes são indispensáveis para reduzir a exploração do espaço de estados e privilegiar as regiões promissoras durante a resolução de um DCOP.

### 2.2.1 Definição de um DCOP

Formalmente, um DCOP é composto pela tupla  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$ , onde: i)  $\mathcal{A}$  representa o conjunto de agentes do sistema; ii)  $\mathcal{X}$  refere-se ao conjunto de variáveis distribuídas entre os agentes em  $\mathcal{A}$ ; iii)  $\mathcal{D}$  representa o conjunto de domínios das variáveis em  $\mathcal{X}$ ; e iv)  $\mathcal{R}$  representa o conjunto as restrições para variáveis em  $\mathcal{X}$ . Cada variável  $x \in \mathcal{X}$  é mantida por um único agente e possui um domínio discreto e finito  $D \in \mathcal{D}$ . Por sua vez, cada valor  $d \in D$  representa um possível estado de um agente  $a \in \mathcal{A}$ . Já as restrições são denotadas por um conjunto  $\mathcal{R}$  de funções de custo ou recompensa sobre um subconjunto de atribuição das variáveis em  $\mathcal{X}$  (YEOH; YOKOO, 2012).

Uma função de custo  $f_{ij} \in \mathcal{R}$  envolvendo um par de variáveis  $x_i$  e  $x_j$  é dada por  $f_{ij} : D_i \times D_j \rightarrow N$  (MODI et al., 2005). Em outras palavras, estas funções expressam um custo ou recompensa sobre cada combinação de valores para as variáveis envolvidas na restrição. Dois agentes  $a_i$  e  $a_j$  são denominados vizinhos quando existir alguma restrição envolvendo as variáveis mantidas por estes agentes (LESSER; ORTIZ; TAMBE, 2003).

Uma solução  $A$  representa as atribuições de um subconjunto de variáveis em  $\mathcal{X}$  (YEOH; FELNER; KOENIG, 2010). Complementarmente, uma solução  $A$  é denominada completa se possuir atribuições para todas as variáveis do problema, isto é,  $\forall x \in \mathcal{X}$ . Desta maneira, a resolução de um DCOP consiste em encontrar uma solução completa onde  $A^* = \{d_1, \dots, d_n \mid d_1 \in D_1, \dots, d_n \in D_n\}$ , de modo que o custo acumulado  $F$  seja minimizado, conforme

$$A^* = \underset{A \in \mathcal{S}}{\operatorname{argmin}} F(A) = \sum_{x_i, x_j \in \mathcal{X}} f_{ij}(d_i, d_j) \quad (2.1)$$

quando  $x_i \leftarrow d_i, x_j \leftarrow d_j$  em  $A$  (MAILLER; LESSER, 2004).

A Equação 2.1 descreve a função objetivo global de um DCOP, onde  $\mathcal{S}$  representa o espaço de estados, isto é, o conjunto de todas as possíveis soluções completas para um determinado problema. Por sua vez, cada função de custo  $f_{ij} \in \mathcal{R}$  representa um fator na função objetivo global. Deste modo, um DCOP requer sua modelagem por meio de técnicas de programação por restrição, a qual constitui a base deste formalismo.

### 2.2.2 Programação por Restrição

A programação por restrição, *Constraint Programming* (CP), refere-se ao estudo de sistemas computacionais baseados em restrições, sendo um dos inúmeros métodos de resolução de problemas da Inteligência Artificial (ROSSI; BEEK; WALSH, 2006). Em termos gerais, a programação por restrição objetiva a resolução de problemas por meio de restrições definidas sobre a natureza da aplicação. Nas últimas décadas este paradigma tem ganhado muita atenção devido ao seu potencial para modelar naturalmente diversas classes de problemas do mundo real.

Em sua essência, uma restrição refere-se a uma relação lógica entre uma ou mais variáveis, onde cada variável possui um domínio predefinido. Deste modo, o objetivo destas relações lógicas consiste em restringir possíveis valores que as variáveis podem assumir. As restrições podem especificar parte das informações de um dado problema, podem ser heterogêneas, isto é, restrições entre variáveis de domínios diferentes, são declarativas, aditivas e raramente são independentes (BARTÁK, 2001).

O CP dispõe de um conjunto de técnicas para resolução de diversas classificações de problemas, em especial problemas envolvendo escalonamento de recursos, planejamento, coordenação ou otimização combinatória. Uma das técnicas mais populares do CP é conhecida como Problema de Satisfação de Restrição (TSANG, 1993). Nesta técnica, o objetivo consiste em encontrar um conjunto de atribuições para cada variável de modo que nenhuma restrição seja violada.

### 2.2.3 Problema de Satisfação de Restrição

Um Problema de Satisfação de Restrição, ou *Constraint Satisfaction Problem* (CSP), visa encontrar uma combinação de ações a fim de alcançar um estado alvo que inicialmente é desconhecido (TSANG, 1993). Contudo, métodos inteligentes ou heurísticas podem ser aplicadas durante a resolução destes problemas de modo a acelerar a exploração do espaço de estados durante a busca (WEISS, 1999).

Um CSP pode ser descrito como um problema de decisão que envolve a atribuição de valores para um conjunto de variáveis, considerando um conjunto de restrições que descrevem as relações entre tais variáveis. Deste modo, um CSP é composto por um conjunto de variáveis, um conjunto de domínios discretos e finitos para cada variável e um conjunto de restrições aplicadas sobre os possíveis valores (TSANG, 1993). Uma solução para um CSP somente é válida quando todas as variáveis possuírem uma atribuição, sem que nenhuma restrição seja violada.

### 2.2.4 Problema de Satisfação Distribuída de Restrição

O CSP é um formalismo para resolução centralizada de problemas combinatórios complexos (TSANG, 1993). No entanto, a formulação clássica do CSP pode ser estendida para atuar de forma distribuída, sendo assim aplicável em sistemas multiagente. Inúmeros problemas da IAD podem ser modelados como Problema de Satisfação Distribuída de Restrição, *Distributed Constraint Satisfaction Problem* (DisCSP).

Em um DisCSP, as variáveis do problema são distribuídas entre agentes, cujo objetivo consiste em coordenar as ações de cada agente a fim de evitar ótimos locais (YOKOO et al., 1998). Em outras palavras, um ótimo local refere-se à uma solução global não ótima, porém, não há nenhuma ação local futura capaz de satisfazer as restrições violadas (WEISS, 1999).

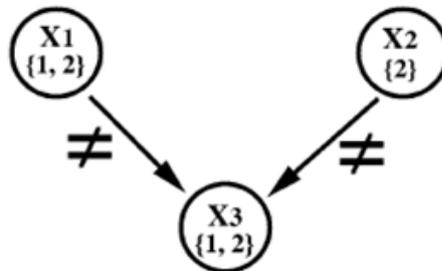


Figura 2.1: Exemplo de um DisCSP representado por meio de um grafo de restrições. Adaptada de (YOKOO et al., 1998).

Um DisCSP é tipicamente representado por meio de um grafo de restrições. Cada agente corresponde a um vértice e as arestas representam as restrições compartilhadas entre pares de vértices conectados (Figura 2.1). O problema é resolvido por meio de computações locais realizadas sobre cada vértice do grafo. Por sua vez, a ordem de execução destas computações locais pode ser arbitrária, predefinida ou altamente flexível, bem como envolver operações síncronas ou assíncronas (YOKOO et al., 1998).

### 2.2.5 Problema de Otimização de Restrição

Estratégias baseadas em satisfação de restrição são capazes de encontrar soluções válidas, quando existentes. No entanto, este formalismo não é capaz de modelar problemas onde as soluções estão associadas a um grau de qualidade (YOKOO et al., 1998). Esta característica deve-se ao fato de problemas de satisfação de restrição considerar apenas restrições com produtos binários (ROSSI; BEEK; WALSH, 2006). Em outras palavras, um CSP é limitado no sentido de que somente é possível designar as soluções encontradas como satisfatórias ou insatisfatórias.

Um Problema de Otimização de Restrição, ou *Constraint Optimization Problem* (COP), é uma generalização do CSP onde as restrições do problema estão associadas a custos ou recompensas. Em outras palavras, cada restrição do problema é caracterizada como uma função de otimização, também chamada de função de custo ou recompensa. Portanto, um COP consiste em encontrar um conjunto de atribuições para cada variável de modo que o custo ou recompensa global da solução seja otimizado. O custo ou recompensa global também conhecido como utilidade da solução (PETCU; FALTINGS, 2005a). Em contrapartida, a exploração do espaço de estados em um COP normalmente é maior e mais complexa do que em um CSP (YOKOO et al., 1998).

### 2.2.6 Problema de Otimização Distribuída de Restrição

O Problema de Otimização Distribuída de Restrição, ou *Distributed Constraint Optimization Problem* (DCOP), apesar de ser uma generalização natural do DisCSP, oferece técnicas que vão além da busca por soluções satisfatórias ou de simples métodos de otimização (LESSER; ORTIZ; TAMBE, 2003). Em um DCOP, assim como em um DisCSP, as variáveis também são distribuídas entre agentes. Contudo, as computações locais em um DCOP visam otimizar o custo ou recompensa da função objetivo global definida para o problema.

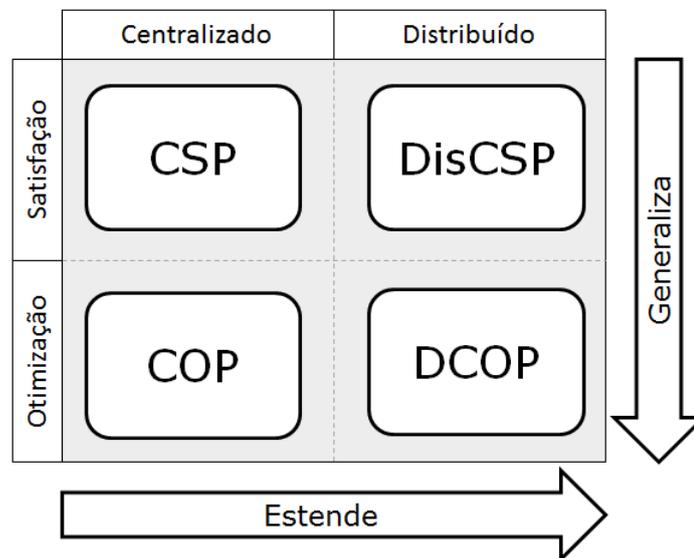


Figura 2.2: Relação entre problemas de satisfação de restrição e problemas de otimização de restrição.

A Figura 2.2 ilustra a relação existente entre problemas de satisfação de restrição e problemas de otimização de restrição. Um DCOP é, portanto, uma generalização de um CSP para lidar com problemas de otimização envolvendo um grau de qualidade

nas soluções encontradas, ao mesmo tempo, uma extensão desta abordagem clássica e centralizada para atuar de maneira distribuída (YEOH; YOKOO, 2012).

Outra característica presente em um DCOP é a eficiência na exploração de estados candidatos quando a qualidade da solução estiver dentro de um limite mínimo predefinido. Este recurso é útil quando o ambiente do problema em questão apresenta restrições de tempo ou de esforço computacional, no entanto, a qualidade da solução aproximada estará dentro de uma distância previamente estipulada (MODI et al., 2005). Ambientes complexos em larga escala ou com características de tempo real são candidatos para aplicação de tais abordagens incompletas (FARINELLI et al., 2008).

## 2.3 Representação de um DCOP

O modelo de representação de um DCOP é fundamental para assegurar uma coordenação eficiente dos agentes. Em termos gerais, a representação está estritamente relacionada com a sistemática que cada algoritmo emprega para a resolução de um DCOP (LEITE; ENEMBRECK; BARTHÈS, 2014). As informações que compõem a representação de um DCOP envolvem: i) domínios de cada variável; ii) variáveis que são controladas em cada agente; iii) vizinhos de cada agente; e iv) funções de custo envolvendo subconjuntos de variáveis.

Outro aspecto relevante sobre a representação de um DCOP refere-se a privacidade das informações, cujo propósito compreende em assegurar que o conhecimento local dos agentes seja protegido durante a resolução do problema (GREENSTADT et al., 2006). Algumas aplicações de sistemas multiagente, como por exemplo, agendamento de reuniões e assistentes pessoais (MAHESWARAN et al., 2006), podem requerer técnicas específicas para preservar a privacidade das informações dos agentes por meio da adição de variáveis auxiliares na definição do problema.

### 2.3.1 Grafo de restrições

Um DCOP é comumente representado estruturalmente por meio de um grafo não dirigido  $G(\mathcal{V}, \mathcal{E})$ , onde os vértices  $\mathcal{V} = \{1, \dots, N\}$  correspondem às variáveis e as arestas  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  entre pares de vértices correspondem às restrições do problema. Este formato de representação permite que os agentes realizem buscas ou inferências a partir de um contexto local, considerando as restrições entre os agentes vizinhos (YOKOO et al., 1998). Além disto, esta representação torna possível a computação paralela em regiões distintas do grafo, permitindo uma resolução local e distribuída dos subproblemas.

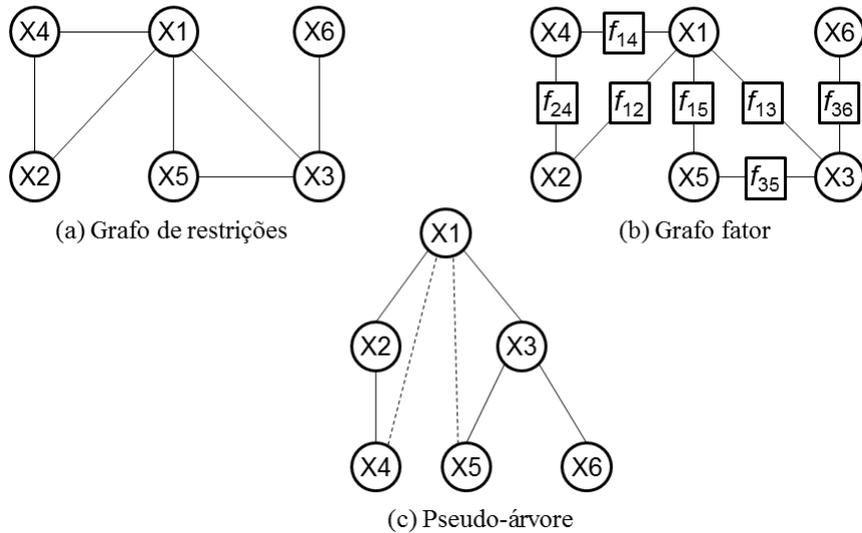


Figura 2.3: Representação de um DCOP por meio de grafos.

Neste sentido, a teoria de grafos (WASSERMAN; FAUST, 1994) fornece diversas propriedades e conceitos frequentemente utilizados para descrever um DCOP. A densidade do grafo, por exemplo, é uma medida que indica a quantidade de arestas existentes. Em outras palavras, grafos com alta densidade possuem uma grande quantidade de arestas, ao passo que grafos com poucas arestas possuem baixa densidade, conhecidos também como grafos esparsos. Outra propriedade relevante é a topologia da rede, a qual descreve o padrão de conexão entre os vértices de um grafo.

A Figura 2.3 ilustra um exemplo de um DCOP representado por meio de três grafos distintos. O grafo de restrições (Figura 2.3.a) é a maneira mais usual para representar um DCOP, entretanto, é necessário que todas as restrições entre as variáveis sejam binárias. Ou seja, para representar um DCOP por meio de um grafo, as funções de custo devem envolver, no máximo, duas variáveis. Do contrário, as restrições  $n$ -árias onde  $n > 2$  devem ser decompostas em restrições binárias (LESSER; ORTIZ; TAMBE, 2003).

### 2.3.2 Grafo fator

Uma maneira alternativa para descrever um DCOP consiste em representá-lo a partir de um grafo fator. Um grafo fator corresponde a um grafo bipartido composto por arestas não direcionais e dois conjuntos disjuntos de vértices, onde cada lado da aresta conecta um vértice de cada conjunto. Na representação de um DCOP a partir de um grafo fator, um conjunto de vértices corresponde às variáveis do problema e o outro conjunto representa as funções de custo (ROGERS et al., 2011).

A Figura 2.3 ilustra um DCOP representado a partir de um grafo fator. Neste

exemplo, o DCOP representado como um grafo fator (Figura 2.3.b) possui as mesmas características e propriedades do grafo de restrições (Figura 2.3.a), porém, destaca-se os dois conjuntos de vértices para representar separadamente as variáveis e restrições do problema. Uma das vantagens da utilização de grafo fator é possibilidade de modelar restrições  $n$ -árias, considerando valores arbitrários para  $n$ .

Outro benefício também presente em grafos fatores é a representação explícita das dependências existentes entre as variáveis e restrições do problema. Em outras palavras, as funções de custo devem estar diretamente conectadas apenas com as variáveis envolvidas na restrição, ao passo que as variáveis devem estar diretamente conectadas apenas com as funções de custo impactadas pela variável em questão. Esta forma de representação é geralmente utilizada em algoritmos para DCOP baseados em passagem de mensagem (FARINELLI et al., 2008; ROGERS et al., 2011).

### 2.3.3 Pseudoárvores

Alguns algoritmos para DCOP são incapazes de lidar com ciclos no grafo de restrições ou podem requerer uma ordenação prévia das variáveis durante o processo de resolução do problema. Um ciclo é um caminho fechado que contém exatamente dois nós iguais, sendo eles o primeiro e o último (WASSERMAN; FAUST, 1994). Uma das abordagens para eliminar ciclos e definir uma ordem de prioridade das variáveis na própria representação de um DCOP consiste em transformar o grafo de restrições em uma pseudoárvore (ALI; KOENIG; TAMBE, 2005).

Por definição, uma pseudoárvore é uma árvore de dispersão onde cada vértice pode estar conectado a outros vértices de maior prioridade na hierarquia. Contudo, apenas um dos vértices de maior prioridade é definido como pai, enquanto os demais vértices de maior prioridade são denominados pseudopai. Além disto, quando um vértice está conectado a um pseudopai, este vértice é chamado de pseudofilho (PETCU; FALTINGS, 2005a).

A Figura 2.3 apresenta um DCOP representado por meio de uma pseudoárvore. No exemplo, a pseudoárvore (Figura 2.3.c) representa o mesmo DCOP do grafo de restrições (Figura 2.3.a), cujas arestas pontilhadas indicam a relação de pseudopai e pseudofilho entre pares de vértices conectados. Uma técnica amplamente utilizada para geração de pseudoárvores consiste em realizar uma busca em profundidade sobre o grafo de restrições (MODI et al., 2005). No entanto, a profundidade ou a largura da pseudoárvore podem influenciar significativamente no desempenho do processo de exploração.

Não obstante, produzir uma pseudoárvore ótima em termos de balanceamento dos vértices é um problema NP-completo (PETCU; FALTINGS, 2005a). Em virtude disto,

diversas pesquisas introduziram técnicas bem-sucedidas para melhorar o desempenho de algoritmos para DCOP baseados em pseudoárvore. Os pesquisadores Ali, Koenig e Tambe (ALI; KOENIG; TAMBE, 2005) introduziram técnicas de pré-processamento baseadas em programação dinâmica para acelerar o processo de busca do algoritmo ADOPT. Já o trabalho de Leite et al. (LEITE et al., 2010) utilizou medidas de centralidade e prestígio oriundas da análise de redes sociais, com o propósito de privilegiar os agentes com maior relevância sobre o problema na hierarquia da pseudoárvore.

## 2.4 Variações de DCOP

Em sua essência, a formulação de um DCOP prevê agentes cooperativos, com comportamentos e conhecimentos determinísticos (YEOH; YOKOO, 2012). Além disso, o ambiente é estático e também determinístico. No entanto, algumas categorias de problemas reais possuem particularidades que dificultam ou impedem a utilização desta formulação clássica de um DCOP. Em outras situações diferenciadas, o próprio objetivo do problema pode ser diferente de uma formulação tradicional de um DCOP. Para lidar com estas limitações e ampliar a aplicabilidade desta técnica de coordenação descentralizada, algumas variações notáveis do formalismo DCOP foram introduzidas na literatura. A seguir serão apresentadas as principais variações de DCOP introduzidas até então.

### 2.4.1 DCOP Quantificado

Em um DCOP tradicional os agentes devem cooperar para otimizar a soma de seus custos locais. No entanto, alguns agentes podem escolher valores para suas variáveis sem nenhum tipo de cooperação. Em casos especiais, tais agentes podem atribuir valores que resultem na pior qualidade global da solução devido às visões parciais sobre o problema. Neste sentido, DCOP Quantificado, *Quantified DCOP* (Q-DCOP), introduz o uso de quantificadores universal e existencial,  $\forall$  e  $\exists$ , para distinguir as variáveis entre cooperativas ou não cooperativas (MATSUI et al., 2010).

A partir da definição original de um CSP, um CSP Quantificado, *Quantified CSP* (Q-CSP), determina uma lista de variáveis quantificadas na forma  $Q.C = q_0x_0, \dots, q_nx_n.C$ , onde  $Q$  corresponde as variáveis e  $q_i$  indica o quantificador existencial  $\exists$  ou universal  $\forall$  para cada variável  $x_i$  (GENT et al., 2008). A definição de um Q-CSP é recursiva e é descrita como: i) se  $C$  é vazio,  $Q.C$  é verdadeiro; ii) se  $Q$  está na forma  $\exists x_0q_1x_1\dots q_nx_n$ , então  $Q.C$  é verdadeiro caso houver um valor  $d \in D_0$  tal que  $q_1x_1\dots q_nx_n.(C \cup x_0 = d)$  é verdadeiro; e iii) se  $Q$  está na forma  $\forall x_0q_1x_1\dots q_nx_n$ , então  $Q.C$  é verdadeiro caso  $q_1x_1\dots q_nx_n.(C \cup x_0 = d)$

for verdadeiro para todos os valores de  $d \in D_0$  (MATSUI et al., 2010).

Por sua vez, um Q-DCOP determina uma sequência de variáveis quantificadas, similarmente ao Q-CSP. Um Q-DCOP é representado por  $Q.(C, F) = q_0x_0, \dots, q_nx_n.(C, F)$ , onde  $Q$  corresponde a sequência de variáveis e  $q_i$  denota o quantificador existencial  $\exists$  ou universal  $\forall$  para cada variável  $x_i$ . Em termos gerais, o objetivo do Q-DCOP é encontrar uma solução global ótima para o DCOP correspondente (MATSUI et al., 2010).

Entretanto, para alcançar tal objetivo, a definição do problema deve ser alterada devido aos quantificadores universais e existenciais. Variáveis existenciais correspondem às variáveis não cooperativas, ao passo que as variáveis universais podem assumir qualquer valor, isto é, representam variáveis cooperativas. Desta maneira, a solução ótima em um Q-DCOP pode ser diferente de um DCOP. Um Q-DCOP define limites superiores e inferiores da solução ótima (*upper-bound* e *lower-bound*) enquanto um DCOP resulta em uma solução completa de custo ótimo (MATSUI et al., 2010).

Em outras palavras, a melhor escolha em um Q-DCOP define o menor limite inferior (*LB*). Por outro lado, no pior caso as variáveis universais podem assumir valores que aumentam o custo da solução candidata. Portanto, o pior caso define o menor limite superior (*UB*). Estas propriedades de um Q-DCOP também são observadas na teoria dos jogos, cujo propósito consiste em indicar quais elementos do problema são cooperativos e quais são competitivos.

#### 2.4.2 DCOP Assimétrico

Um DCOP pode apresentar restrições que produzem diferentes recompensas ou custos entre os agentes envolvidos. Recompensas ou funções de custo assimétricas entre agentes não podem ser representadas naturalmente por meio de um DCOP tradicional (GRUBSHEIN et al., 2010). Embora alguns modelos de representação sejam capazes de descrever recompensas ou funções de custo assimétricas a partir de um DCOP tradicional, tais técnicas elevam a complexidade do problema devido a adição de restrições ou variáveis privadas (MAHESWARAN et al., 2004).

Em virtude disto, o trabalho de Grinshpoun et al. (GRINSHPOUN et al., 2013) introduziu uma nova formulação para lidar com recompensas assimétricas em um DCOP, denominada *Asymmetric Distributed Constraint Optimization Problem* (A-DCOP). O conceito chave em um A-DCOP refere-se à analogia entre a formulação de um DCOP e uma categoria de jogo da teoria de jogos conhecida como *Potential Games*. A importância desta analogia está relacionada ao fato de que cada jogo potencial finito possui, no mínimo, uma estratégia de Equilíbrio de Nash pura, ou *Nash Equilibrium* (NE).

Na teoria dos jogos, uma estratégia NE pura representa um perfil estável das ações correspondentes considerando todos os participantes, onde qualquer alteração unilateral realizada por um único agente não irá impactar em uma maior recompensa local para o participante. Em comparação à formulação tradicional de um DCOP, tal definição da estratégia NE pura está relacionada com soluções conhecidas como ótimo local, seja mínimo ou máximo. Estas soluções representam um conjunto de atribuições realizadas entre todos os agentes, onde uma única mudança na atribuição por um agente irá reduzir apenas o custo global, isto é, o custo local permanecerá o mesmo.

Em um A-DCOP, cada restrição  $C$  aplicada sobre as variáveis do  $k$ -ésimo agente é representada por meio de um mapeamento entre os domínios das variáveis, onde este mapeamento resulta em um único valor real  $C : D_{i1} \times \dots \times D_{ik} \rightarrow \mathbb{R}^+ \cup \{0\}$ . Esta definição significa que o custo ou recompensa de uma restrição é o mesmo para todos os agentes participantes. Como consequência, quando um agente reduz seu custo em uma restrição, todas as restrições envolvidas irão compartilhar um decremento semelhante no custo desta restrição (GRINSHPOUN et al., 2013).

Desta maneira, ao considerar um ótimo local, qualquer mudança nas atribuições de uma variável poderá apenas reduzir ambos os custos global e local. O propósito do A-DCOP consiste em, portanto, representar a assimetria combinatória em um DCOP quando o problema em questão prevê agentes com diferentes recompensas ou custos em uma mesma restrição compartilhada.

### 2.4.3 DCOP Multi-Objetivo

Em um DCOP típico, a função objetivo do problema representa um único objetivo global de otimização. No entanto, esta definição torna-se um desafio quando a natureza do problema a ser modelado requer uma função de otimização multi-objetivo entre os agentes. Outra dificuldade deste cenário reside em como promover a privacidade dos objetivos individuais dos agentes. Neste sentido, diversas técnicas foram propostas para lidar naturalmente com DCOP multi-objetivo, ou *Multi-Objective DCOP* (MO-DCOP) (BOWRING; TAMBE; YOKOO, 2006; MATSUI et al., 2008; OKIMOTO et al., 2014).

Uma das técnicas mais notáveis para lidar com MO-DCOP consiste em introduzir camadas adicionais de restrições por recursos (BOWRING; TAMBE; YOKOO, 2006). Para tal, é necessário prever uma nova categoria de função de custo  $g_i$  no subconjunto das restrições de  $x_i$ . Deve existir também um limite para o custo de um determinado recurso, denominado *g-budget*. Portanto, o custo  $G_i$  de um recurso deve ser menor ou igual ao *g-budget* deste recurso. Como consequência, uma *g-function* aliada ao *g-budget* constitui

uma restrição por recurso denominada *g-constraint*.

Contudo, esta técnica de representação multi-objetivo requer a revisão do grafo de restrições para adicionar variáveis virtuais que detêm as *g-constraint*  $n$ -ária. As variáveis virtuais indicam as violações de restrições de recursos, realizando podas no espaço de busca sempre que o custo exceder o limite predefinido. Esta representação parte do princípio que as *g-function* não podem ser tratadas como *f-function*, no entanto, cada atribuição de valor deve considerar ambas as categorias de funções. Desta forma, uma importante característica desta abordagem de DCOP multi-objetivo é a interação entre as *f-function* e as *g-function* para preempção do espaço de estados do problema (MATSUI et al., 2008).

Formalmente, um MO-DCOP com restrições de recursos pode ser estendido a partir de um DCOP clássico conforme

$$F(A) = \sum_{x_i, x_j \in \mathcal{X}} f_{ij}(d_i, d_j) \quad (2.2)$$

$$\forall x_i \in \mathcal{X}: g_i(d_i, \{d_j \mid x_j \in \text{vizinhos}(x_i)\}) \leq G_i$$

quando  $x_i \leftarrow d_i$  e  $x_j \leftarrow d_j$  em  $A$  (BOWRING; TAMBE; YOKOO, 2006). Por fim, os aspectos mais relevantes sobre um MO-DCOP envolvem a dominância das funções entre as diferentes classes de restrições e a garantia de atribuições ótimas que satisfaçam todas as restrições, também conhecida como eficiência de Pareto (FAVE et al., 2011).

#### 2.4.4 DCOP Dinâmico

Conforme mencionado anteriormente, um DCOP típico prevê definições estruturais estáticas sobre o problema e o comportamento dos agentes. Não obstante, é bastante comum em problemas reais a necessidade de adaptação em resposta às mudanças no ambiente ao longo do tempo. Um DCOP dinâmico, ou *Dynamic DCOP* (D-DCOP), estende a formulação padrão  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} \rangle$  de um DCOP, permitindo mudanças na definição do problema em cada tempo  $t$  dado por  $D_t = \langle \mathcal{A}^t, \mathcal{X}^t, \mathcal{D}^t, \mathcal{R}^t \rangle$ , considerando um dado intervalo  $1 \leq t \leq T$ .

De fato, um dos maiores desafios em um D-DCOP consiste em encontrar a melhor solução para cada sequência de DCOP em um intervalo de tempo por meio da resolução de múltiplos DCOP (HOANG et al., 2016). Este desafio eleva ainda mais a complexidade de resolução do problema. Em função disto, uma característica muito importante para garantir uma rápida convergência da solução em cada sequência de DCOP reside na capacidade de autoestabilização dos agentes durante a exploração dos estados candidatos (PETCU; FALTINGS, 2005b).

Para abordar este desafio, diversos métodos para D-DCOP têm sido investigados (PETCU; FALTINGS, 2005b; YEOH et al., 2011; PETCU; FALTINGS, 2007b). Grande parte destes estudos objetivam acelerar o processo de estabilização em resposta a uma mudança no ambiente. Recentemente Hoang et al. (HOANG et al., 2016) propuseram um novo método de resolução de DCOP para considerar alterações futuras no ambiente através do uso de abordagens preditivas durante a resolução do problema, denominado D-DCOP proativo, ou *Proactive D-DCOP* (PD-DCOP).

#### 2.4.5 DCOP Probabilístico

Em sua essência, um DCOP prevê uma definição determinística do problema. Não obstante, também é comum problemas reais cujos ambientes possuem características não determinísticas. Esta particularidade ocorre, sobretudo, devido a ocorrência de eventos aleatórios que podem impactar diretamente sobre o resultado das ações de cada agente. Diante disto, diversos pesquisadores têm recentemente investido esforços em técnicas para resolução de DCOP probabilístico, *Probabilistic DCOP* (P-DCOP), introduzindo mecanismos para lidar com a incerteza sobre os custos ou recompensas do problema (NGUYEN; YEOH; LAU, 2012; WU; JENNINGS, 2014; LE et al., 2016).

O trabalho de Nguyen, Yeoh e Lau (NGUYEN; YEOH; LAU, 2012) introduziu uma variação de DCOP, denominada DCOP estocástico, ou *Stochastic DCOP* (S-DCOP), cujas restrições definem funções de distribuição de probabilidade de recompensa. Por sua vez, os pesquisadores Wu e Jennings (WU; JENNINGS, 2014) estenderam esta definição para prever a adição de funções de custo ou recompensas estocásticas em um DCOP, denominado *Uncertain Reward DCOP* (UR-DCOP).

Por definição, funções de custo estocásticas adicionam dois novos componentes em um DCOP tradicional, formando a tupla  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R}, \mathcal{E}, \mathcal{S} \rangle$ , onde  $\mathcal{E} = \{s_1, \dots, s_m\}$  refere-se à um conjunto de variáveis aleatórias  $\forall r_i \in \mathcal{R}$  e  $\mathcal{S} = \{S_1, \dots, S_m\}$  representa um conjunto de domínios finitos  $s_j \in S_j$  sobre cada variável aleatória em  $\mathcal{E}$ , respectivamente (WU; JENNINGS, 2014). Desta forma, as funções de custo devem considerar tanto as variáveis de decisão quanto as variáveis aleatórias para calcular o custo ou recompensa em uma dada solução parcial (GHOSH; KUMAR; VARAKANTHAM, 2015).

Em um UR-DCOP, a distribuição de probabilidade de uma dada variável aleatória  $s_i$  com domínio  $S_i$  é descrita como a crença de uma variável aleatória e denotada por  $b_i \in \Delta(S_i)$ . Por sua vez,  $\mathbf{b} = \langle b_1, \dots, b_m \rangle$  representa a crença conjunta de todas as variáveis aleatórias. Quando  $\mathbf{b}$  é conhecido, a resolução do UR-DCOP consiste em encontrar um conjunto  $A$  de atribuições que minimize a soma das utilidades das funções de custo.

Quando  $\mathbf{b}$  é desconhecido, é necessário considerar múltiplas crenças conjuntas contidas em um determinado espaço de crenças  $\mathcal{B}$ . Em (LE et al., 2016), foi proposta uma simplificação desta formulação, denominada *Expected Regret DCOP* (ER-DCOP), onde a crença de uma variável aleatória pode ser dependente dos valores das variáveis de decisão.

## 2.5 Algoritmos para DCOP

De maneira geral, os métodos para resolução de DCOP podem ser agrupados em categorias como síncronos ou assíncronos, completos ou incompletos, parcialmente centralizado ou totalmente distribuído, dentre outras. Estas categorizações referem-se às estratégias de resolução empregadas por tais métodos de resolução que, em geral, estão diretamente associadas a questões como a qualidade das soluções encontradas, tempo de resposta, privacidade da informação ou escalabilidade do algoritmo.

Yeoh (YEOH, 2010) propôs uma taxonomia para classificação dos algoritmos para DCOP. A Figura 2.4 apresenta uma extensão desta taxonomia, onde foram introduzidas novas categorias entre os principais algoritmos para DCOP encontrados na literatura até o momento. As informações incorporadas nesta extensão foram: i) se o algoritmo possui um processo síncrono ou assíncrono; ii) se o algoritmo dispõe de mecanismos para suporte à garantia de qualidade; e iii) estratégia baseada em amostragem.

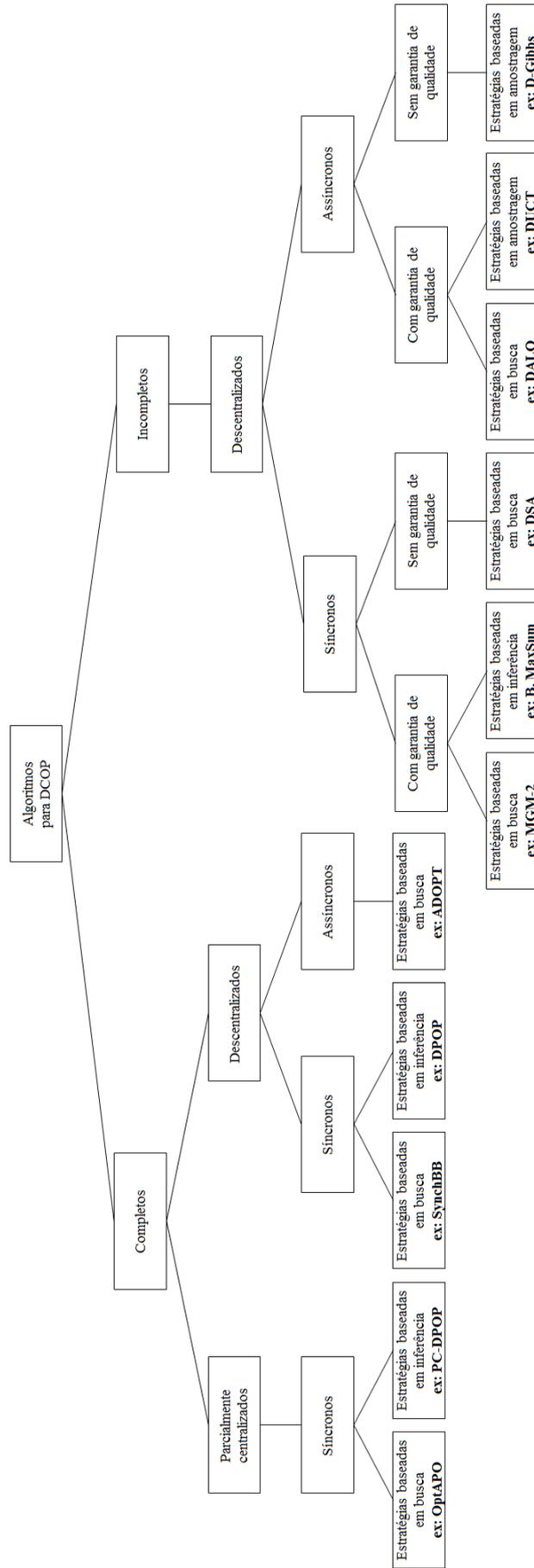


Figura 2.4: Categorias dos algoritmos para DCOP.

### 2.5.1 Algoritmos Baseados em Busca, Inferência e Amostragem

Os algoritmos para DCOP podem ser classificados em três estratégias distintas para exploração do espaço de estados do problema: busca, inferência e amostragem. Em estratégias baseadas em busca, os agentes percorrem o grafo de restrições de maneira distribuída, visto que esta busca é guiada pelas heurísticas introduzidas no algoritmo. Os métodos mais utilizados para exploração dos estados em algoritmos baseados em busca são: *best-first* (MODI et al., 2005), *depth-first* (CHECHETKA; SYCARA, 2006), *backtracking* (GERSHMAN; MEISELS; ZIVAN, 2009) e *branch-and-bound* (YEOH, 2010).

Outras abordagens tipicamente utilizadas em algoritmos para DCOP referem-se a estratégias baseadas em inferência (YEOH, 2010). Tais estratégias empregam técnicas de programação dinâmica ou propagação de crenças para computar o custo agregado de um agente, considerando os custos de seus vizinhos (ROGERS et al., 2011). Por sua vez, estes custos agregados são propagados para os agentes vizinhos de maneira sistemática, reduzindo o tamanho do problema a cada etapa do algoritmo. Adicionalmente, é comum a utilização de estratégias baseadas em inferência em algoritmos de passagem de mensagem (GHOSH; KUMAR; VARAKANTHAM, 2015; RUST; PICARD; RAMPARANY, 2016).

Algumas pesquisas recentes introduziram novos algoritmos incompletos baseados em amostragem estatística (NGUYEN; YEOH; LAU, 2013; FIORETTO et al., 2014). Estas estratégias são provenientes da teoria de probabilidade e utilizam amostras do espaço de estados para criar funções aproximadas de distribuição de probabilidade a partir de inferências estatísticas. Os algoritmos baseados em amostragem normalmente requerem um parâmetro inicial para definição do intervalo de confiança aceitável para as amostras inferidas (OTTENS; DIMITRAKAKIS; FALTINGS, 2012).

### 2.5.2 Algoritmos Completos e Incompletos

Algoritmos completos compreendem os métodos de resolução capazes de encontrar soluções ótimas para um DCOP, quando existirem (PETCU, 2007). Contudo, em alguns cenários reais, sobretudo aplicações em larga escala, o uso de tais estratégias pode ser impraticável em função do esforço computacional requerido pelo processo de resolução do problema. Algoritmos completos, em geral, requerem recursos computacionais de ordem exponencial em relação à complexidade do problema, o que pode limitar o uso de estratégias completas (PEARCE; TAMBE; MAHESWARAN, 2008). A complexidade do problema pode ser impactada pela topologia ou densidade do grafo de restrições, quantidade de variáveis ou pelo tamanho dos domínios das variáveis.

Em contrapartida, algoritmos incompletos fornecem soluções aproximadas, isto é,

subótimas, contudo, geralmente requerem um esforço computacional menor em relação aos algoritmos completos (PETCU, 2007). Devido a este fato, diversos pesquisadores têm concentrado esforços no desenvolvimento de novos algoritmos incompletos, os quais são mais favoráveis para adoção em ambientes complexos reais, tais como larga escala, tempo real, severamente limitados em termos de recursos computacionais, ambientes dinâmicos ou incertos (PETCU, 2007; ROGERS et al., 2011; KIEKINTVELD et al., 2010).

Adicionalmente, alguns algoritmos incompletos são capazes de oferecer garantia de qualidade das soluções subótimas encontradas dentro de um intervalo predefinido. Em outras palavras, tais algoritmos incompletos oferecem mecanismos de aproximação de erro que possibilitam a convergência da qualidade de solução para um limite mínimo especificado (PEARCE; TAMBE; MAHESWARAN, 2008).

Outra característica notável presente em alguns algoritmos incompletos para DCOP é a capacidade de retornar a melhor solução válida encontrada até o momento quando o processo de resolução é interrompido, antes do término do algoritmo. Esta propriedade, denominada *anytime*, objetiva aprimorar gradativamente a qualidade das soluções em virtude do tempo de execução do algoritmo (YEOH et al., 2011). Para tal, a propriedade *anytime* é assegurada por um mecanismo ou uma função de convergência monotônica (ZIVAN; OKAMOTO; PELED, 2014).

### 2.5.3 Algoritmos Parcialmente Centralizados e Distribuídos

Os algoritmos para DCOP são, em sua essência, distribuídos. No entanto, algumas estratégias resolvem os conflitos locais através de uma centralização parcial do problema. Esta estratégia previne que os agentes tomem decisões que resultem em um máximo local, evitando conflitos desnecessários (PETCU; FALTINGS; MAILLER, 2007). Nesta abordagem, um agente resolve de maneira centralizada parte do problema e, em seguida, notifica todos os agentes envolvidos no subproblema sobre a solução encontrada (MAILLER; LESSER, 2004).

Não obstante, tais métodos de resolução parcialmente centralizados aumentam consideravelmente a perda de privacidade de informações, visto que o agente que centraliza a resolução dos conflitos precisa conhecer o domínio e as funções de custo dos demais agentes envolvidos no subproblema (GREENSTADT et al., 2006). Por outro lado, os algoritmos totalmente distribuídos reduzem esta perda de privacidade, contudo, requerem maior esforço computacional para resolução de conflitos (MAILLER; LESSER, 2004). Este aumento do esforço computacional é comumente notado no volume de mensagens necessárias para estabilizar uma determinada solução parcial do problema.

#### 2.5.4 Algoritmos Síncronos e Assíncronos

Algoritmos para DCOP objetivam a eficiência através do processamento paralelo e distribuído. Em algoritmos assíncronos, um agente é capaz de tomar suas decisões com base apenas em sua visão local do problema, isto é, sem depender de alguma mensagem específica de seus vizinhos para realizar processamentos locais (MODI et al., 2005). Este é um requisito fundamental em aplicações em larga escala, pois esta característica garante que os agentes atuem completamente independente durante o processo de resolução do problema. Contudo, esta característica requer um processo de busca mais complexo, bem como um modelo de comunicação mais sofisticado.

Em contrapartida, apesar dos algoritmos assíncronos reduzirem o tempo ocioso dos agentes, não há garantia sobre a consistência das visões locais dos agentes. Como consequência, os agentes podem tomar decisões a partir de uma visão inconsistente e ter que revisá-las futuramente (PERI; MEISELS, 2013). Já os algoritmos síncronos implementam um método de exploração sistemático dividido em etapas ou fases bem definidas, cujos agentes devem aguardar o recebimento de uma mensagem particular para desempenharem suas ações (PETCU; FALTINGS, 2005a). Este comportamento síncrono aumenta o tempo ocioso dos agentes, mas por outro lado, garante visões locais consistentes dos agentes sobre o processo exploratório das soluções candidatas.

Neste sentido, Peri e Meisels (PERI; MEISELS, 2013) abordaram a relação entre diferentes níveis de sincronização em algoritmos para DCOP. O estudo demonstrou que visões inconsistentes são mais prejudiciais do que a ociosidade dos agentes em relação ao tempo total e desempenho do processo de resolução, sobretudo justificado pelo esforço requerido pelos agentes revisarem suas ações. Portanto, alguns algoritmos apresentam um melhor desempenho ao introduzir algum nível de sincronização no processo de resolução do problema (NETZER; GRUBSHTEIN; MEISELS, 2012).

## 2.6 Aplicações de DCOP

O DCOP tem se mostrado uma técnica importante para raciocínio distribuído, problemas envolvendo planejamento e decisão e coordenação descentralizada em sistemas multiagente (LEITE; ENEMBRECK; BARTHÈS, 2014). Cenários naturalmente complexos como planejamento, negociação, agendamento de eventos, escalonamento de recursos e sincronização são exemplos candidatos para aplicação deste formalismo. A seguir são apresentados alguns trabalhos relevantes sobre aplicações de DCOP.

### 2.6.1 Coloração de grafos

Problemas de coloração de grafo são tipicamente utilizados como um meio para avaliação de desempenho e comparação entre algoritmos para DCOP. O problema de coloração de grafos consiste em selecionar uma cor para cada vértice do grafo, de maneira que nenhum vértice adjacente compartilhe a mesma cor. Além disto, a coloração dos vértices deve utilizar a menor quantidade possível de cores. Na formulação deste problema como um DCOP, os valores de cada variável representam possíveis cores e as restrições tem por objetivo assegurar que todas as variáveis adjacentes assumam cores distintas (LEITE; ENEMBRECK; BARTHÈS, 2014).

### 2.6.2 Redes de Sensores, Wireless e Smart-Grids

Um problema do mundo real frequentemente formulado como um DCOP refere-se à coordenação em redes de sensores (LESSER; ORTIZ; TAMBE, 2003). Dentre as possíveis aplicações de redes de sensores, destacam-se o rastreamento e determinação de movimentos de um ou mais alvos situados no escopo da rede (Figura 2.5). Devido ao fato de cada sensor possuir um alcance de rastreamento limitado, um modelo de coordenação entre os sensores vizinhos é necessário para determinar o movimento dos alvos dentro do espaço de cobertura da rede.

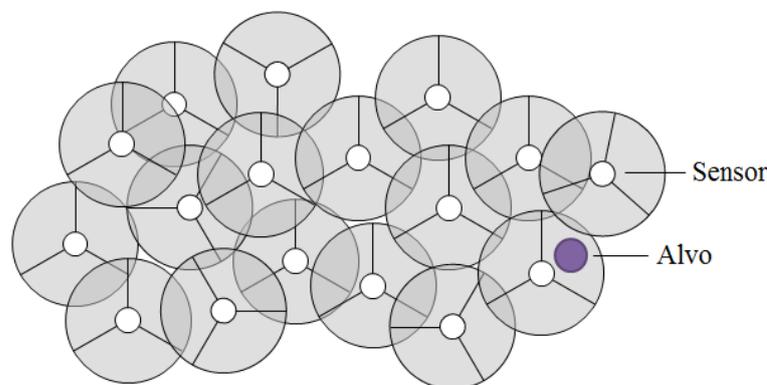


Figura 2.5: Exemplo de uma rede de sensores.

Problemas de coordenação em redes de sensores são retratados de forma distribuída devido à escalabilidade do ambiente, escassez de recursos computacionais dos dispositivos e redução de pontos de falhas. Adicionalmente, alguns estudos endereçaram cenários de configuração, auto-organização e coordenação em redes wireless formulados como um DCOP (FARINELLI et al., 2008; MONTEIRO et al., 2012; MULDOON et al., 2013). Estes estudos demonstraram a eficiência de alguns algoritmos para DCOP em termos

de custo de comunicação entre os dispositivos wireless, sincronização temporal de dados compartilhados e colaboração para propagação das mensagens.

Alguns trabalhos recentes retrataram a aplicação do formalismo DCOP como modelo de coordenação em smart-grids (AGRAWAL; KUMAR; VARAKANTHAM, 2015; CERQUIDES; PICARD; RODRÍGUEZ-AGUILAR, 2015). O modelo de coordenação visa, em especial, a auto-organização dos dispositivos da rede de distribuição a fim de garantir um fluxo contínuo de energia conforme a capacidade de carga da malha elétrica. Os dispositivos são classificados como consumidores, produtores ou ambos, os quais representam as variáveis em um DCOP. Já as restrições visam assegurar um fluxo de energia suficiente para atender a demanda dos dispositivos consumidores, contudo, respeitando a capacidade de carga da rede.

### 2.6.3 Agendamento de Eventos e Alocação de Recursos

Outras categorias de problemas reais formuladas como um DCOP são cenários envolvendo agendamento de eventos e alocação de recursos compartilhados. Maheswaran et al. (MAHESWARAN et al., 2004) propuseram modelos genéricos de DCOP para estas categorias de problemas. Nos modelos propostos, as variáveis do problema representam janelas de tempo ou eventos e as restrições referem-se às funções de custo relacionadas às atividades em conjunto ou recursos compartilhados. Sobre o agendamento de eventos, destaca-se a dificuldade em garantir a privacidade das informações da agenda de cada participante durante a resolução do problema.

Sobre problemas envolvendo alocação distribuída de recursos, Carpenter et al. (CARPENTER et al., 2007) introduziram uma abordagem para evacuação de desastres como um DCOP. Na formulação definida para o problema, as variáveis são as rotas de evacuação ou destinos e as restrições representam as funções de custo considerando o tempo de viagem, capacidade dos abrigos e interdição de rotas. O objetivo do modelo proposto consiste em encontrar rotas de evacuação adequadas em cenários dinâmicos.

Recentemente pesquisadores propuseram um sistema para controle do suprimento de água formulado como um MO-DCOP (AMIGONI; CASTELLETTI; GIULIANI, 2015), cujo principal desafio refere-se às restrições conjuntas de diferentes recursos como cidades, barragens e fazendas. Esta categoria de problema, tipicamente referenciada na literatura como DCOP com restrições de recurso, *Resource Constrained* (RC-DCOP), possibilita a representação de restrições denominadas *soft* ou *hard* (MATSUI et al., 2008). Restrições *hard* invalidam uma solução candidata quando violadas. Em contrapartida, restrições *soft* são utilizadas sobretudo para otimização, isto é, não invalidam a solução quando violadas.

#### 2.6.4 Sincronização, Planejamento e Coordenação de Ações

O trabalho de Jungles e Bazzan (JUNGES; BAZZAN, 2008) abordou o problema de sincronização distribuída de semáforos por meio de um DCOP. Na modelagem proposta para a sincronização de semáforos, as variáveis representam cruzamentos entre as estradas e as restrições definem funções de custo sobre os planos de sinais em cada cruzamento. O objetivo global do problema consiste em encontrar o melhor plano de sinais entre os cruzamentos de modo a minimizar o tempo de parada em semáforos e proporcionar um sistema progressivo de tráfego.

Leite, Giacomet e Enembreck (LEITE; GIACOMET; ENEMBRECK, 2009) propuseram um modelo de planejamento para condução automática de composições férreas como um DCOP. As variáveis do problema denotam janelas de tempo para um conjunto de ações e as restrições representam o impacto de cada ação sobre as ações futuras. A função objetivo global do modelo proposto visa encontrar um conjunto de ações que minimizam o consumo de combustível e o tempo de viagem, além de considerar na escolha das ações questões relacionadas à segurança na condução.

Em outro trabalho, Yedidsion e Zivan (YEDIDSION; ZIVAN, 2016) introduziram uma modelagem genérica para coordenar as ações de uma equipe de robôs móveis como um DCOP. No cenário abordado, os robôs possuem câmeras com ângulo de visão limitado e precisam coordenar suas ações para atingir seus objetivos individuais, tais como seguir ou rastrear um determinado objeto. No entanto, por se tratar de robôs móveis, a configuração do ambiente é alterada constantemente.

## 2.7 Revisão Crítica

O formalismo DCOP tem emergido como uma elegante técnica para coordenação descentralizada em sistemas multiagente. Este formalismo oferece soluções para diversos desafios essencialmente presentes em um sistema multiagente, tais como a privacidade de informação, autonomia dos agentes, raciocínio distribuído e coordenação de ações. Estes aspectos não são encontrados em métodos centralizados, o que reforça a importância e o potencial do DCOP como mecanismo para coordenação descentralizada em sistemas multiagente (LEITE; ENEMBRECK; BARTHÈS, 2014).

Apesar de ser uma linha de pesquisa recente, o DCOP tem despertado grande interesse dos pesquisadores em função da capacidade de modelar naturalmente diversas classes de problemas do mundo real. Como consequência, inúmeros algoritmos para DCOP têm sido propostos até então. Os algoritmos implementam diferentes estratégias para

resolução, as quais em alguns casos podem ser restritivas devido às particularidades de um problema em questão (PERI; MEISELS, 2013). Portanto, em função das características de cada problema, a escolha do algoritmo para DCOP é fundamental para obter um desempenho satisfatório e atender adequadamente as restrições do ambiente.

### 2.7.1 Evolução dos Algoritmos

Historicamente os primeiros algoritmos propostos para DCOP são variantes ou possuem alguma influência dos métodos de resolução já conhecidos para CSP e DisCSP. Tal influência é justificada pelo fato do DCOP ser uma generalização natural do DisCSP. A principal diferença entre estes dois formalismos encontra-se na definição da função objetivo do problema, no entanto, esta diferença eleva significativamente a complexidade em um DCOP. A seguir são apresentados resumidamente os principais algoritmos para DCOP disponíveis na literatura até o momento.

#### 2.7.1.1 Algoritmos Completos

**SynchBB** (HIRAYAMA; YOKOO, 1997): o algoritmo *Synchronous Branch-and-Bound* (SynchBB) foi o primeiro método de resolução completo proposto para DCOP. O algoritmo implementa uma estratégia *branch-and-bound* que guia a busca pelo grafo de restrições por meio de um custo global acumulado, chamado *bound*. Uma das limitações do algoritmo SynchBB está associada ao comportamento síncrono dos agentes, devido a dependência de uma informação globalmente atualizada durante a busca.

**ADOPT** (MODI et al., 2005): o algoritmo *Asynchronous Distributed Constraint Optimization* (ADOPT) foi o primeiro método completo e assíncrono proposto para DCOP. O ADOPT consolida a essência de um sistema multiagente no que diz respeito à autonomia dos agentes e privacidade de informação. Antes de iniciar a busca, o ADOPT requer uma ordenação dos agentes em uma estrutura de pseudoárvore. Em seguida, os agentes realizam uma busca oportunista *best-first* sobre a pseudoárvore que representa o problema. No entanto, alcançar um estado estável no ADOPT normalmente exige um número elevado de mensagens em função do crescimento ocorrer em ordem exponencial, no pior caso.

**Variações do ADOPT**: após o ADOPT, diversas variações deste algoritmo foram propostas para endereçar algumas ineficiências. O ADOPT-ng (SILAGHI; YOKOO, 2006) introduz um novo tipo de mensagem, chamado *nogood*, para eliminar a necessidade de pré-processamento do grafo de restrições e acelerar a resolução de conflitos. Por sua

vez, o BnB-ADOPT (YEOH; FELNER; KOENIG, 2010) introduz uma estratégia de busca *branch-and-bound* para reduzir as reconstruções de soluções abandonadas devido à busca oportunista da versão original. O MCA (BOWRING; TAMBE; YOKOO, 2006) é uma extensão do ADOPT para endereçar MO-DCOP. O DJAO (KIM; LESSER, 2014) combina técnicas do ADOPT e grafos E/OU visando uma exploração mais eficiente do espaço de estados do problema. Por fim, recentemente foi proposta uma versão híbrida do ADOPT, chamada BD-ADOPT, a qual combina as estratégias de busca *best-first* e *depth-first* (CHEN et al., 2017).

**OptAPO** (MAILLER; LESSER, 2004): o *Optimal Asynchronous Partial Overlay* (OptAPO) implementa uma busca parcialmente centralizada para acelerar a estabilização de ações conflitantes em um subgrupo de agentes. Neste caso, quando detectada alguma situação de conflito, os agentes envolvidos formam uma sessão de mediação para realizar a resolução de forma centralizada. Não obstante, os agentes precisam compartilhar seus conhecimentos e visões locais para o agente mediador possa escolher a melhor solução parcial para a sessão de mediação corrente. Este aspecto eleva a perda da privacidade das informações locais dos agentes.

**DPOP** (PETCU; FALTINGS, 2005a): o *Dynamic Programming Optimization Protocol* (DPOP) utiliza uma estratégia baseada em inferência, cujo propósito consiste em computar e propagar os custos agregados de cada variável. Estes custos agregados são calculados por meio de técnicas da programação dinâmica. Uma das vantagens do DPOP é o número linear de mensagens requeridas para execução do algoritmo, contudo, o tamanho das mensagens apresenta um crescimento exponencial, no pior caso.

**Variações do DPOP:** várias extensões do DPOP foram propostas para melhorar o desempenho em situações específicas. O MB-DPOP (PETCU; FALTINGS, 2007a) é um algoritmo *memory-bounded* para endereçar o crescimento exponencial das mensagens do DPOP original, porém, nesta versão o volume de mensagens não é linear. Por sua vez, o PC-DPOP (PETCU; FALTINGS; MAILLER, 2007) utiliza uma técnica de centralização parcial para evitar também o crescimento exponencial das mensagens, no pior caso. Finalmente, o S-DPOP (PETCU; FALTINGS, 2005b) aborda questões de autoestabilização dos agentes em ambientes dinâmicos.

**NCBB** (CHECHETKA; SYCARA, 2006): o *No-Commitment Branch-and-Bound* (NCBB) é baseado em uma estratégia *branch-and-bound* síncrona. O NCBB, assim como outros algoritmos para DCOP, também requer que os agentes estejam ordenados em uma estrutura de pseudoárvore. Uma das principais características deste algoritmo é o aprendizado sobre as atribuições dos vizinhos predecessores.

**AFB** (GERSHMAN; MEISELS; ZIVAN, 2009): no *Asynchronous Forward Boun-*

*ding* (AFB), os agentes realizam a propagação de soluções parciais assincronamente e executam *backtracking* sempre que o custo exceder o limite superior atual (*upper-bound*). Para assegurar a consistência das visões locais durante a busca assíncrona, os agentes utilizam uma marcação temporal (*timestamp*) anexada em cada solução parcial. Além disto, o AFB possui uma extensão para A-DCOP, denominada *Asymmetric Two-Way Bounding* (ATWB) (GRINSHPOUN et al., 2013).

**ConcFB** (NETZER; GRUBSHTEIN; MEISELS, 2012): o algoritmo *Concurrent Forward Bounding* (ConcFB) introduz uma abordagem alternativa, onde o espaço de estados é dividido em múltiplos processos de busca concorrentes em regiões disjuntas do problema. Como resultado, cada processo produz uma solução parcial para um subespaço de estados. Desta forma, o ConcFB cria dinamicamente novos processos de busca ao detectar novos subespaços promissores.

Apesar de diversos algoritmos completos terem sido propostos até então, todos demandam um custo computacional de ordem exponencial em alguma condição, seja em função da quantidade de ciclos de mensagem, tamanho das mensagens ou da quantidade de vezes que uma restrição é avaliada em cada agente. Aspectos como a privacidade da informação são, por vezes, prejudicados para melhorar o desempenho em algoritmos completos. Não obstante, em alguns cenários reais, tais como agendamento de eventos, é fundamental manter a privacidade de informação, o que inviabiliza o uso de algoritmos com esta característica.

### 2.7.1.2 Algoritmos Incompletos

**DBA e DSA** (ZHANG et al., 2005): o *Distributed Breakout Algorithm* (DBA) e o *Distributed Stochastic Algorithm* (DSA) são os algoritmos incompletos mais populares para DCOP, cuja ideia chave está centrada na busca local. O DBA e o DSA são algoritmos *anytime* e são comumente utilizados como base em algoritmos mais recentes. No DBA, os agentes alteram os custos das restrições na tentativa de evitar máximos locais. Já o DSA introduz uma abordagem não-determinística para escolha local das ações dos agentes através de uma probabilidade  $p$ . Além disto, o DSA possui uma versão para A-DCOP, denominada *Asymmetric Coordinated Local Search* (ACLS) (GRINSHPOUN et al., 2013).

**MGM** (MAHESWARAN; PEARCE; TAMBE, 2004): o *Maximum Gain Message* (MGM) é uma modificação do DBA, cuja diferença reside no fato de que o MGM não altera o custo da restrição para evitar um máximo local. Assim, no MGM os agentes propagam os valores selecionados entre os vizinhos. Em seguida, cada agente poderá alterar seus valores apenas quando resultar em recompensas unilaterais. Com isto, caso

a recompensa local for maior que a recompensa máxima de todos os agentes vizinhos, o agente atribui o novo valor em sua variável. Assim como o DSA, o MGM também possui uma variante para A-DCOP, chamada *Guaranteed Convergence Asymmetric MGM* (GCA-MGM) (GRINSHPOUN et al., 2013).

**MGM-2** e **SCA-2** (PEARCE; TAMBE; MAHESWARAN, 2008): os algoritmos *Maximum Gain Message-2* (MGM-2) e *Stochastic Coordination Algorithm-2* (SCA-2) são extensões do MGM e DSA para oferecer suporte a garantia de qualidade das soluções encontradas. Os algoritmos MGM-2 e SCA-2 são baseados no conceito  $k$ -ótimo, o qual consiste na formação de grupos de agentes onde nenhum grupo com  $k$  ou menos agentes é capaz de melhorar a solução atual. Neste caso, são garantidas soluções subótimas cujo limite de qualidade estará associado a uma distância  $k = 2$ .

**DALO** (KIEKINTVELD et al., 2010): o algoritmo *Distributed Asynchronous Local Optimization* (DALO) oferece como principal inovação a utilização dos conceitos  $k$ -ótimo ou  $t$ -distância como mecanismos para coordenação local envolvendo grupos de agentes. Tais mecanismos evitam que os agentes realizem atribuições visando ganhos exclusivamente locais. O conceito  $t$ -distância refere-se a uma distância máxima  $t$  de um agente central entre os demais agentes do grupo. A abordagem  $k$ -ótimo é mais recomendada em grafos esparsos, em função da quantidade máxima de agentes em um grupo ser limitada. Já a abordagem  $t$ -distância é sugerida em grafos com alta densidade devido à quantidade máxima de grupos ser limitada.

**min-max**, **beta-alpha** e **bi-threshold ADOPT** (MATSUI et al., 2010): estas extensões do ADOPT têm por objetivo endereçar Q-DCOP. O objetivo dos agentes em um Q-DCOP consiste em calcular o *lower-bound* e *upper-bound* para uma solução através de quantificadores universais e existenciais. As variáveis universais podem receber qualquer valor e não serão atribuídas durante o processo de otimização. Já as variáveis existenciais devem assumir exatamente um dos possíveis valores. Estes algoritmos implementam estratégias de decisão populares, tais como *min-max*, *beta-alpha* e *bi-threshold*.

**Bounded Max-Sum** (ROGERS et al., 2011): o Bounded Max-Sum objetiva o relaxamento do problema por meio da eliminação dos ciclos no grafo de restrições. Os ciclos são removidos através da eliminação de dependências entre as funções de custo e variáveis. Esta eliminação é realizada considerando as dependências que resultem no menor impacto na qualidade da solução. Em seguida, o grafo de restrições é transformado em uma pseudoárvore e os agentes realizam uma inferência sobre esta nova estrutura utilizando uma técnica de propagação de crença conhecida como distribuição marginal.

**DaCSA** (VINYALS et al., 2010): o algoritmo *Divide and Coordinate Subgradient Algorithm* (DaCSA) consiste em dividir o DCOP em subproblemas menores para serem

resolvidos localmente e coordenar as ações dos agentes para alcançarem um consenso dos subproblemas. Para tal, o DaCSA utiliza métodos de relaxação dual lagrangiana e subgradiente para decompor o problema. Além disto, o DaCSA oferece suporte à garantia de qualidade das soluções subótimas.

**DUCT** (OTTENS; DIMITRAKAKIS; FALTINGS, 2012): o *Distributed Upper Confidence Tree* (DUCT) introduz uma nova abordagem para resolução de DCOP baseada em técnicas da teoria da probabilidade e intervalos de confiança. O DUCT utiliza amostras do subespaço de estados para computar a distribuição de probabilidade conjunta entre os estados dentro de um determinado intervalo de confiança  $\epsilon$ . No entanto, o DUCT apresenta um crescimento exponencial de memória em relação a quantidade de agentes envolvidos no problema.

**D-Gibbs** (NGUYEN; YEOH; LAU, 2013): o *Distributed Gibbs* (D-Gibbs) é um algoritmo *memory-bounded* também baseado em intervalos de confiança. Uma amostra Gibbs é uma cadeia de Markov que pode ser utilizada para calcular aproximações de distribuição de probabilidade conjunta. Por sua vez, o D-Gibbs realiza uma inferência distribuída para obter a distribuição de probabilidade conjunta do problema dada uma quantidade suficiente de amostras.

**LS-RAND** (HOANG et al., 2016): o *Local Search Random* (LS-RAND) é um algoritmo *anytime* de busca local para ambientes dinâmicos, onde as funções de custo podem mudar durante o processo de resolução do problema. Alguns algoritmos para DCOP tradicional são capazes de lidar com D-DCOP, no entanto, tais abordagens reagem à mudança no problema e, em seguida, resolvem a versão atual do problema. O LS-RAND resolve esta deficiência atuando proativamente na resolução de problemas com base em um horizonte de mudanças, reutilizando as informações obtidas de versões anteriores do problema para acelerar a convergência da solução da versão atual do problema. Para tal, o algoritmo conta com variáveis de decisão e variáveis aleatórias para controlar eventos estocásticos que podem causar mudanças no ambiente.

Embora diversos métodos incompletos sejam capazes de garantir um nível mínimo de qualidade sobre as soluções encontradas, ainda permanecem em aberto questões sobre como definir este limite. Como consequência, um algoritmo incompleto pode produzir soluções inadequadas para um determinado problema, mesmo oferecendo mecanismos para garantia da qualidade das soluções. Em outros casos, um algoritmo incompleto pode apresentar um desempenho similar a um algoritmo completo em função da complexidade do problema. Outro desafio inerente à garantia de qualidade da solução refere-se ao fato da definição do limite ser intimamente relacionado a um problema em particular.

### 2.7.2 Comparação dos Algoritmos

Em geral, os algoritmos para DCOP derivam de métodos de resolução bastante conhecidos e explorados na literatura. Em virtude disto, os algoritmos não dispõem de uma fundamentação teórica diversificada, isto é, os métodos são baseados, sobretudo, em técnicas de programação por restrição, teoria dos jogos ou teoria da probabilidade. Neste sentido, as diferenças entre os algoritmos concentram-se essencialmente nas estratégias de resolução ou heurísticas empregadas.

A Tabela 2.1 apresenta uma classificação, bem como uma comparação entre os principais algoritmos para DCOP disponíveis na literatura. Os critérios considerados para comparação são: i) categoria dos algoritmos; ii) se o algoritmo possui um comportamento parcialmente síncrono ou totalmente assíncrono; iii) se o algoritmo possui uma abordagem de resolução parcialmente centralizada ou totalmente distribuída; iv) fundamentação teórica utilizada pelo algoritmo; e v) métricas de avaliação e o protocolo utilizado para avaliação de desempenho com outros algoritmos.

De maneira geral, os experimentos realizados pelos autores dos algoritmos não introduzem variações na topologia dos grafos de restrições. Com isto, grande parte das avaliações dos algoritmos limitou-se a poucas categorias de problemas reais. Assim, análises mais aprofundadas envolvendo uma maior diversidade de topologias de grafos são recomendadas para determinar com maior rigor as condições em que os algoritmos são favoráveis, com base nas características do problema em questão.

Outra questão sobre os experimentos realizados pelos autores dos algoritmos está relacionado ao tamanho dos problemas envolvidos. Alguns ambientes reais, como por exemplo redes de sensores, tipicamente envolvem centenas de variáveis ou restrições (LESSER; ORTIZ; TAMBE, 2003). Contudo, nos experimentos reportados na literatura, os algoritmos foram avaliados sobretudo a partir de problemas envolvendo poucas dezenas de agentes ou restrições. Além disto, não há análises direcionadas para escalabilidade dos algoritmos, o que é um aspecto fundamental ao se tratar de problemas ou ambientes complexos e em larga escala.

Com relação às métricas de avaliação empregadas nos experimentos, na maioria dos casos são considerados ciclos de mensagens e o número de verificação de restrições (NCCC) para algoritmos completos e ciclos de mensagens e qualidade das soluções encontradas para algoritmos incompletos. Ciclos de mensagens consideram apenas o tempo de computação de um dado agente ao receber uma mensagem (MODI et al., 2005), ao passo que o NCCC aumenta à medida que os agentes computam o impacto de cada restrição ao receber uma mensagem (MEISELS et al., 2002).

Tabela 2.1: Avaliação experimental dos algoritmos para DCOP.

Algoritmo	Estratégia	Fundamentação	Características	Avaliação	Métricas	Comparativo
ADOPT	<i>Best-first</i>	Programação por restrição	- Completo - Assíncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 3 - Domínio igual a 3 - Número de variáveis igual a 26	- Ciclos de mensagens	SynchBB
DPOP	Propagação de custo	Programação dinâmica	- Completo - Síncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Número de variáveis igual a 200 - Número de restrições igual a 341	- Ciclos de mensagens	ADOPT
OptAPO	Mediação cooperativa	Programação por restrição	- Completo - Síncrono - Parcialmente centralizado	- Grafos aleatórios - Densidade igual a 3 - Domínio igual a 3 - Número de variáveis igual a 26	- Ciclos de mensagens	ADOPT
NCBB	<i>Branch and bound</i>	Programação por restrição	- Completo - Síncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 3 - Domínio igual a 3 - Número de variáveis igual a 26	- Ciclos de mensagens	ADOPT DPOP
AFB	<i>Branch and bound</i>	Programação por restrição	- Completo - Assíncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Prob. conexão igual a 98% - Domínio igual a 3 - Número de variáveis igual a 10	- NCCC - Número de mensagens	SynchBB ADOPT
ConcFB	<i>Branch and bound</i>	Programação por restrição	- Completo - Síncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 5 - Domínio igual a 3 - Número de variáveis igual a 12	- NCCC - Número de mensagens	SynchBB BnB-ADOPT ODPOP
MGM-2/3 SCA-2/3	<i>k</i> -ótimo	Teoria dos jogos	- Incompleto - Síncrono - Descentralizado	- Grafos aleatórios - Densidade igual a 5 - Domínio igual a 3 - Número de variáveis igual a 1000	- Ciclos de mensagens	MGM DSA
Bounded Max-Sum	Lei distributiva generalizada	Programação dinâmica	- Incompleto - Síncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 3 - Domínio igual a 3 - Número de variáveis igual a 50	- Ciclos de mensagens - Qualidade da solução	Max-Sum
DALO- <i>k</i> DALO- <i>t</i>	<i>k</i> -ótimo <i>t</i> -distância	Programação por restrição	- Incompleto - Assíncrono - Distribuído - Pré-processamento	- Grafos aleatórios - Densidade igual a 4 - Domínio igual a 10 - Número de variáveis igual a 100	- Ciclos de mensagens - Qualidade da solução	KOPT
min-max ADOPT alpha-beta ADOPT bi-threshold ADOPT	Variáveis quantificadas	Teoria dos jogos	- Incompleto - Assíncrono - Parcialmente centralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 2 - Domínio igual a 10 - Número de agentes igual a 10	- Ciclos de mensagens	-
ACLS GCA-MGM	Custos assimétricos	Teoria dos jogos	- Incompleto - Síncrono - Distribuído - Pré-processamento	- Grafos aleatórios - Densidade igual a 2 - Domínio igual a 10 - Número de variáveis igual a 10	- Ciclos de mensagens - Qualidade da solução - Perda de privacidade	MGM DSA
DaCSA	Dividir e coordenar	Lagrange subgradiente	- Incompleto - Síncrono - Descentralizado	- Topologias aleatória, regular e mundos pequenos - Domínio igual a 2 - Número de variáveis igual a 49	- Ciclos de mensagens - Qualidade da solução	DSA Max-Sum
DUCT	Intervalos de confiança	Teoria da probabilidade	- Incompleto - Assíncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Densidade igual a 3 - Domínio igual a 5 - Número de variáveis igual a 26	- Tempo de execução - Qualidade da solução	DPOP DSA MGM MGM-2
D-Gibbs	Intervalos de confiança	Teoria da probabilidade	- Incompleto - Assíncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Prob. de conexão igual a 100% - Domínio igual a 20 - Número de variáveis igual a 30	- Tempo de execução - Qualidade da solução	DUCT DPOP MGM MGM-2
LS-RAND	Variáveis aleatórias	Teoria dos jogos	- Incompleto - Síncrono - Descentralizado - Pré-processamento	- Grafos aleatórios - Prob. troca custo igual a 50% - Domínio igual a 3 - Número de variáveis igual a 12	- Tempo de execução - Qualidade da solução	S-DPOP

Não obstante, alguns cenários do mundo real necessitam manter a privacidade das informações dos agentes, tais como agendamento de eventos. Portanto, a perda da privacidade de informação poderia ser mensurada nos experimentos realizados. Outras métricas que também poderiam ser consideradas nos experimentos referem-se ao tamanho máximo das mensagens e volume total de informações em memória para cada agente. Um sensor, por exemplo, possui uma grande limitação acerca da capacidade de processamento, memória e comunicação. Portanto, a falta de métricas apropriadas para avaliação dos algoritmos resulta em experimentos parciais, no sentido de abranger poucas categorias de problemas reais.

### 2.7.3 Aplicabilidade dos Algoritmos

Algoritmos para DCOP requerem esforço computacional de ordem exponencial em relação a complexidade do problema, mas nem sempre esta informação está evidente nas avaliações experimentais dos algoritmos. Além disto, ambientes reais normalmente possuem particularidades que limitam a utilização de alguns algoritmos. Os algoritmos completos, por exemplo, são indicados em cenários que necessitam de soluções ótimas, envolvem poucos agentes e quando o grafo de restrições é esparso. Em contrapartida, os algoritmos incompletos são recomendados em ambientes em larga escala, quando a aplicação possui restrição de tempo ou quando há severas restrições computacionais.

O ADOPT e suas variantes, por exemplo, requerem uma grande quantidade de mensagens durante o processo de resolução do problema. Portanto, o ADOPT não é recomendado em cenários cujo ambiente possua uma rede não confiável. Outros algoritmos como NCBB, AFB e ConcFB também requerem um número exponencial de mensagens no pior caso. Contudo, estes algoritmos reduzem significativamente o número médio de mensagens introduzindo processos síncronos na busca. Por sua vez, o DPOP demanda memória em ordem exponencial com relação ao número de agentes para projetar os custos agregados de cada variável. Por fim, a centralização parcial no OptAPO resulta em perda da privacidade de informação.

Em relação aos métodos incompletos, os algoritmos MGM e DSA são capazes de alcançar soluções subótimas mais rapidamente, porém, não há garantia na qualidade das soluções encontradas. Assim, a aplicação destes algoritmos é proibitiva em problemas que requerem um limite mínimo de qualidade das soluções. Já o MGM-2, SCA-2 e o DALO baseiam-se em conceitos da teoria dos jogos para oferecer garantia de qualidade nas soluções. O algoritmo Bounded Max-Sum também possui mecanismos para oferecer uma garantia de qualidade, porém, tal garantia é específica para uma instância em particular

do problema. Por fim, os algoritmos DUCT e D-Gibbs também oferecem garantia de qualidade, entretanto, assim como os demais algoritmos incompletos, a determinação do limite de qualidade não é uma tarefa trivial. A maneira mais comum para determinar o limite mínimo de qualidade baseia-se no pior caso, contudo, ainda assim permanece subjetivo e dependente de uma instância de problema em particular.

## 2.8 Considerações Finais

Em geral, pesquisas recentes sobre DCOP estão centradas no desenvolvimento de novos métodos de resolução. Como resultado, inúmeros algoritmos para DCOP têm sido propostos até então, contudo, tais estudos são limitados pois exploram estratégias bastante conhecidas deste campo de pesquisa. Como consequência, a falta de diversidade teórica resulta em contribuições marginais destes trabalhos, visto que os algoritmos provenientes de uma mesma teoria tipicamente apresentam desempenhos similares. Outra questão em aberto sobre este campo de pesquisa refere-se aos experimentos reportados na literatura, pois se aplicam a poucos cenários reais.

Por fim, estudos direcionados para a aplicação do DCOP em cenários reais, em especial, problemas em larga escala, são fundamentais para reforçar o potencial deste formalismo para modelagem de problemas naturalmente complexos, tais como redes de sensores. Estes cenários geralmente não são favoráveis para os métodos de resolução de DCOP atuais em função da complexidade do problema e da falta de escalabilidade dos algoritmos. Deste modo, no capítulo a seguir será apresentada uma fundamentação teórica alternativa para impulsionar a evolução das pesquisas sobre DCOP.

## Capítulo 3

# Fundamentos sobre Sincronização em Rede de Osciladores Acoplados

Sincronização é um fenômeno físico natural que possibilita descrever a emergência do comportamento coletivo em sistemas dinâmicos complexos. Este fenômeno pode ser observado em diversos sistemas biológicos, tais como a interação em populações de vagalumes (WINFREE, 1967; BUCK; BUCK, 1976), células marca-passo dos sistemas cardiovascular e nervoso (PESKIN, 1975; DYE, 1991), processos químicos e metabólicos (GHOSH; CHANCE; PYE, 1971; SHERMAN; RINZEL; KEIZER, 1988), comportamento coletivo de animais (WALKER, 1969; STROGATZ, 2003), dentre outros. A sincronização corresponde a um dos principais tópicos abordados em sistemas dinâmicos devido à grande aplicabilidade em cenários reais, sobretudo problemas complexos em larga escala.

Este capítulo tem por objetivo fornecer uma introdução sobre sistemas dinâmicos complexos e sincronização física em redes de osciladores acoplados, com ênfase ao modelo matemático proposto por Kuramoto (KURAMOTO, 1984). Neste capítulo também são abordadas algumas variações do modelo de Kuramoto para lidar com cenários envolvendo redes complexas, estabilidade e condições de sincronização, bem como aplicações reais sobre o contexto de sistemas distribuídos.

### 3.1 Sistemas Dinâmicos

Sistemas dinâmicos compreendem a representação e estudo de modelos gerais cujos componentes do sistema evoluem ao longo do tempo seguindo uma determinada regra ou função (LUENBERGER, 1979). Tal evolução do modelo ocorre de maneira iterativa a partir do estado atual dos componentes, produzindo novos estados sucessivamente. Em outras palavras, um sistema dinâmico é naturalmente determinístico e seu estudo tem por objetivo a compreensão da relação entre os estados do sistema, sua trajetória e estabilidade

em um dado intervalo de tempo (PIKOVSKY; ROSENBLUM; KURTHS, 2001).

Um sistema dinâmico pode ser representado como um modelo linear ou não-linear, embora em ambos os cenários estes sistemas são essencialmente determinísticos. Sobre modelos não-lineares, os componentes do sistema evoluem em um intervalo de tempo apresentando um comportamento imprevisível, aleatório e caótico, em função da grande sensibilidade do sistema sobre o estado atual (LUENBERGER, 1979).

A sincronização constitui uma das principais linhas de pesquisa sobre sistemas dinâmicos não-lineares, a qual tem por objetivo investigar a tendência de um dado sistema em operar sincronamente (PIKOVSKY; ROSENBLUM; KURTHS, 2001). Estudos multidisciplinares sobre sincronização têm recebido muita atenção nas últimas décadas, no entanto, este fenômeno é conhecido desde o século XVII.

## 3.2 Modelos Matemáticos de Sincronização

Um dos primeiros pesquisadores a investigar este fenômeno da sincronização foi o físico holandês Christian Huygens, depois de perceber que dois relógios de pêndulo sincronizavam seus ritmos quando apoiados sobre uma mesma base, independentemente do estado inicial dos pêndulos (STROGATZ, 2003). Huygens então deduziu que este comportamento ocorria devido ao acoplamento entre os dois relógios de pêndulo, pois as pulsações de cada pêndulo eram propagadas entre os relógios através da base de apoio compartilhada.

### 3.2.1 Modelo de Winfree

Somente mais tarde, no século XX, este fenômeno da sincronização ganhou mais notoriedade após sua primeira formulação matemática bem-sucedida (STROGATZ, 1997). Mais especificamente, em 1965 o biólogo americano Arthur Winfree iniciou um estudo sobre o comportamento coletivo de uma espécie de vagalumes, em especial, o motivo pelo qual estes insetos emitiam luzes fosforescentes em um ritmo uniforme. Em suas observações, Winfree percebeu que a emissão de luz de cada vagalume estimulava os demais vagalumes vizinhos a regularem coletivamente seus ritmos individuais.

Portanto, Winfree concluiu que a sincronização dos vagalumes ocorre quando tais estímulos compartilhados são suficientes para influenciarem as frequências individuais de cada inseto, ao passo que, ao considerar estímulos fracos, prevalece a anarquia entre eles. Após esta observação, Winfree descreveu tal comportamento dos vagalumes de maneira análoga à uma população de osciladores de fase acoplados (WINFREE, 1967).

Um oscilador é um dispositivo que produz sinais repetitivos autonomamente. Em especial, os osciladores de fase são restritos a amplitudes constantes, cuja fase de cada oscilador representa seu estado em um determinado tempo. O espaço de estados de um oscilador de fase pode ser definido por um intervalo circular  $[0, 2\pi]$ , onde  $0 \equiv 2\pi$ . Quando a fase de um oscilador atinge um certo limite, o oscilador em questão emite um pulso para todos os seus vizinhos. A frequência de tempo em que ocorre cada transmissão de pulso de um oscilador é caracterizada como período.

Deste modo, o comportamento coletivo em um conjunto de osciladores acoplados emerge a partir do ajuste gradual dos períodos sempre que os osciladores receberem novos pulsos de seus vizinhos. Esta dinâmica faz com que os períodos dos osciladores tornem-se cada vez mais semelhantes ao longo do tempo. Assim, Winfree propôs uma formulação matemática para descrever a dinâmica do processo de sincronização (WINFREE, 1967), cuja evolução da fase  $\theta_i$  do oscilador  $i$  é governada por

$$\frac{d\theta_i}{dt} = \omega_i + \left( \sum_{j=1}^N X(\theta_j) \right) Z(\theta_i) \quad (3.1)$$

onde a frequência natural do oscilador  $i$  é caracterizada por  $\omega_i$ . Por sua vez, a função  $X(\theta_j)$  representa a influência de cada oscilador vizinho  $j$ , ao passo que a função  $Z(\theta_i)$  representa a resposta do oscilador  $i$  aos estímulos recebidos dos vizinhos para a fase  $\theta_i$ .

Mais especificamente, cada oscilador está acoplado a um ritmo coletivo produzido pela população de osciladores de maneira análoga à aproximação do campo médio em física (WINFREE, 1967). Sobre a perspectiva da sincronização, a aproximação do campo médio refere-se à tendência dos osciladores ajustarem suas fases para uma fase média.

Winfree também notou que, quando a variação entre as frequências naturais  $\omega_i$  são significativas em relação ao acoplamento  $X(\theta_j)$ , a evolução das fases ocorre de forma incoerente. Este comportamento incoerente permanecerá até ultrapassar um certo limite de acoplamento, cujo limite pode variar de acordo com o grau de divergência entre as frequências naturais dos osciladores (WINFREE, 1967). Em outras palavras, um conjunto de osciladores é passível de sincronização apenas quando há uma força de acoplamento suficiente perante à heterogeneidade das frequências naturais dos osciladores.

### 3.2.2 Modelo de Kuramoto

O modelo matemático introduzido por Winfree para descrever a sincronização física em redes de osciladores acoplados despertou o interesse de diversos pesquisadores sobre o tema, dentre eles, o físico japonês Yoshiki Kuramoto. A partir do trabalho de Winfree,

Kuramoto propôs uma simplificação no modelo para osciladores fracamente acoplados (KURAMOTO, 1975), dado por

$$\frac{d\theta_i}{dt} = \omega_i + \sum_{j=1}^N \Gamma_{ij}(\theta_j - \theta_i) \quad (3.2)$$

cuja função  $\Gamma_{ij}$  representa a interação entre os osciladores  $i$  e  $j$ . No entanto, mesmo após tal simplificação proposta, a análise deste modelo permanecia limitada em cenários com osciladores fracamente acoplados.

Para resolver tal impasse, Kuramoto propôs uma variação deste modelo por meio de um acoplamento senoidal, considerando uma rede de osciladores idênticos ou muito semelhantes globalmente acoplados (KURAMOTO, 1984), conforme

$$\Gamma_{ij}(\theta_j - \theta_i) = \frac{K}{N} \sin(\theta_j - \theta_i) \quad (3.3)$$

onde a constante  $K$  representa a força do acoplamento global e  $N$  representa o número de osciladores da rede. Mais especificamente, o acoplamento deste modelo é descrito pela soma do seno das diferenças das fases para cada par de osciladores acoplados, dado por

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i) \quad (3.4)$$

onde o fator  $\frac{1}{N}$  garante ao modelo um comportamento consistente quando  $N \rightarrow \infty$ . A dinâmica deste modelo conduz os osciladores à uma frequência média  $\Omega$  ao redefinir  $\theta_i \rightarrow \theta_i + \Omega t$  para todo oscilador  $i$ . Assim, para cada oscilador  $i$ , a frequência  $\omega_i$  denota o desvio da frequência média  $\Omega$ .

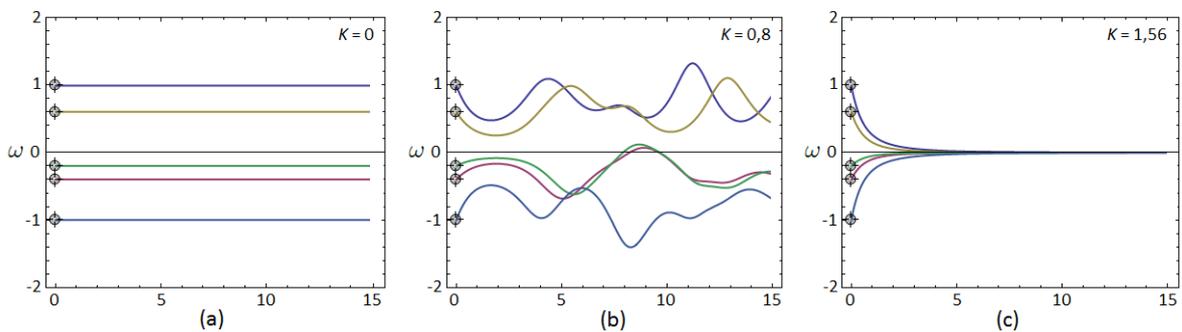


Figura 3.1: Dinâmica de sincronização em uma rede de osciladores globalmente acoplados. Gerado pelo software Mathematica (WOLFRAM, 2003).

A Figura 3.1 ilustra a dinâmica de sincronização descrita pelo modelo de Kuramoto em um conjunto de cinco osciladores globalmente acoplados em três regimes distintos. No

gráfico da Figura 3.1.a, a força de acoplamento é nula, isto é,  $K = 0$ , fazendo com que os osciladores preservem suas frequências naturais  $\omega$  ao longo do tempo. Já no gráfico da Figura 3.1.b, existe uma força de acoplamento  $K = 0,8$  entre os osciladores, contudo, este acoplamento não é suficiente para a ocorrência da sincronização entre os osciladores dado o nível de divergência entre as frequências naturais  $\omega$  dos osciladores. Neste caso, a evolução das fases dos osciladores ocorre de maneira incoerente ao longo do tempo. Por fim, no gráfico da Figura 3.1.c, há uma força de acoplamento  $K = 1,56$  suficiente para a ocorrência da sincronização dos osciladores em relação à divergência entre as frequências naturais  $\omega$  dos osciladores.

Desta maneira, o modelo de Kuramoto descreve a emergência de sincronização em uma rede de osciladores acoplados com comportamentos idênticos ou muito semelhantes, onde a diferença entre os osciladores reside sobretudo na diversidade das frequências naturais  $\omega$ . Ou seja, as frequências naturais dos osciladores estão distribuídas ao longo da rede seguindo uma função de distribuição  $g(\omega)$ , cujo processo de sincronização conduz iterativamente estas frequências  $\omega$  para uma frequência média  $\Omega$ .

### 3.2.3 Parâmetros de Ordem de Kuramoto

Dada uma rede de osciladores acoplados, os parâmetros de ordem  $r(t)$  e  $\psi(t)$  de Kuramoto descrevem o grau de sincronização entre as fases e a fase média respectivamente, onde  $0 \leq r(t) \leq 1$ . Portanto,  $r(t) \approx 1$  indica a coerência entre as fases dos osciladores acoplados, ao passo que  $r(t) \approx 0$  corresponde à evolução das fases sem produzir um ritmo coletivo (KURAMOTO, 1984), conforme

$$re^{i\psi} = \frac{1}{N} \sum_{j=1}^N e^{i\theta_j} \quad (3.5)$$

cujo parâmetro  $\psi$  representa a fase média para todo  $\theta_j$  e  $r$  refere-se à coerência de fase da rede de osciladores em um determinado tempo  $t$ . Assim, os parâmetros de ordem representam o campo médio do sistema dinâmico complexo, o qual permite evidenciar a tendência de convergência e o grau da coerência em uma rede de osciladores acoplados (PIKOVSKY; ROSENBLUM; KURTHS, 2001).

Portanto, ao supor que todas as frequências dos osciladores em uma determinada rede estão diferentes em um determinado momento, então as fases  $\theta_j$  estarão uniformemente distribuídas entre  $[0, 2\pi]$ , indicando assim a ausência do campo médio. Por outro lado, caso alguns osciladores possuam frequências idênticas ou muito similares em um determinado momento, então o campo médio será diferente de zero. Com isto, é possível

reescrever o modelo de Kuramoto de uma maneira mais conveniente para análise da coerência das fases em termos dos parâmetros de ordem, conforme  $\frac{d\theta_i}{dt} = \omega_i + Kr \sin(\psi - \theta_i)$ . Desta maneira, esta representação enfatiza o grau de sincronização em uma rede de osciladores ao longo do tempo.

### 3.2.4 Tipos de Sincronização

O processo de sincronização corresponde à tendência de auto-organização em sistemas dinâmicos. Esta auto-organização pode ser manifestada ou motivada por meio de diferentes regras ou comportamentos, o que pode implicar em diferentes tipos de sincronização. Embora existam diversos tipos de sincronização explorados na literatura, os principais são: sincronização completa, sincronização generalizada, sincronização por atraso e sincronização de fase (PIKOVSKY; ROSENBLUM; KURTHS, 2001).

A sincronização completa é caracterizada pela igualdade dos estados de todos os osciladores do sistema. Este processo ocorre apenas em sistemas onde todos os osciladores são idênticos, isto é, todos os osciladores são regidos pela mesma dinâmica de movimento. Neste cenário, a sincronização completa é observada quando a condição inicial idêntica dos osciladores possibilita uma evolução igualitária do sistema ao decorrer do tempo, permanecendo continuamente inalterada (PIKOVSKY; ROSENBLUM; KURTHS, 2001).

Por sua vez, a sincronização generalizada é observada predominantemente em redes envolvendo osciladores não idênticos, no entanto, com parâmetros iniciais iguais ou ligeiramente diferentes. Este tipo de sincronização ocorre em sistemas dinâmicos complexos denominados mestre-escravo, os quais possuem uma dinâmica que difere da sincronização tradicional por acoplamento externo (PECORA; CARROLL, 1990). Mais especificamente, em redes do tipo mestre-escravo, o pulso de sincronização é gerado a partir de um oscilador denominado mestre para os demais osciladores escravos. Neste caso, prevalece o comportamento dos osciladores mestres sobre o sistema.

Já a sincronização por atraso ocorre entre osciladores não idênticos quando, ao aumentar a força de acoplamento, a transição de fase conduz primeiramente para um estado assíncrono de fase seguido por um estado síncrono. Ou seja, na sincronização por atraso, a transição de fase emerge de maneira deslocada no tempo, onde a dinâmica de um dos osciladores pode seguir ou atrasar as mudanças de estado dos demais osciladores da rede (PIKOVSKY; ROSENBLUM; KURTHS, 2001). Este atraso gera perturbações no processo de sincronização, visto que as interações realizadas em um tempo  $t$  de um determinado oscilador podem ser percebidas somente em um tempo  $t+n$  pelos osciladores vizinhos, onde  $n$  representa um grau de atraso.

Por fim, a sincronização de fase é observada entre osciladores não idênticos quando o acoplamento for suficiente para o sistema convergir para uma frequência comum. Este comportamento conduz as fases dos osciladores para uma fase média  $\psi$ . Portanto, a sincronização de fase ocorre quando a diferença  $\Theta$  entre as fases de todos os osciladores for igual a zero, isto é,  $\Theta = 0$ , condição também conhecida como travamento de fase. Quando os osciladores permanecem em sentidos opostos, isto é,  $\Theta = \pi$ , os osciladores estão em antifase. Finalmente, quando os osciladores possuem uma diferença arbitrária entre as fases, isto é,  $\Theta \neq 0$  e  $\Theta \neq \pi$ , os osciladores encontram-se fora de fase.

Sistemas dinâmicos não idênticos também são passíveis de alcançar um regime intermediário de sincronização de fase, cujas fases estão sincronizadas, entretanto, as amplitudes permanecem diferentes (PIKOVSKY; ROSENBLUM; KURTHS, 2001). Neste cenário, as frequências dos osciladores são idênticas e as fases permanecem as mesmas ao longo do tempo, embora as fases sejam diferentes entre os osciladores.

### 3.2.5 Condição de Sincronização

Estudos sobre condições de sincronização, bem como o intervalo do espaço de estados onde ocorre a sincronização, referem-se a uma subárea de pesquisa sobre sistemas dinâmicos não-lineares amplamente explorada até então. De forma geral, tais estudos concentram-se sobre a perspectiva de sistemas dinâmicos complexos, onde o principal propósito consiste em determinar um limite mínimo necessário da força de acoplamento para que a manifestação de sincronização ocorra em uma rede de osciladores arbitrária (ACEBRÓN et al., 2005).

Uma rede de osciladores acoplados é caracterizada como sincronizada quando todas as frequências  $\omega$  dos osciladores são idênticas ao valor constante da frequência média  $\Omega$  ao longo do tempo (DÖRFLER; M.; F., 2013). Supondo que, em uma determinada rede de osciladores acoplados, as frequências naturais  $\omega$  estão normalizadas e os osciladores deslocam-se sobre uma mesma órbita, a frequência sincronizada da solução será nula, ou seja,  $\Omega = 0$ . Isto ocorre devido ao fato das frequências estarem normalizadas, visto que nesta situação o campo médio sempre será nulo.

Além disto, as fases  $\theta$  são consideradas coesas ao se aproximarem da fase média  $\psi$  ao longo do tempo. Em outras palavras, a coesão das fases ocorre pois  $\theta_i$  tende a se aproximar da fase média  $\psi$  de todos os osciladores, ao invés de se aproximar de uma fase  $\theta$  em particular. Uma solução tem fases coesas quando todos os osciladores acoplados possuem a distância de fase menor que o ângulo  $\gamma \in [0, \frac{\pi}{2}]$ , isto é,  $|\theta_i - \theta_j| \leq \gamma$ , para cada par  $\{i, j\}$  de osciladores acoplados (DÖRFLER; M.; F., 2013).

Todavia, o processo de sincronização está condicionado à força de acoplamento entre os osciladores na rede. Em outras palavras, a sincronização é manifestada quando houver uma força de acoplamento  $K$  suficiente em relação ao acoplamento crítico  $K_c$ , que representa a força de acoplamento mínima necessária para o início da coerência dos osciladores. Assim, dada uma força de acoplamento  $K$ , a sincronização emerge a partir de  $K > K_c = \omega_{max} - \omega_{min}$ , onde  $\omega_{max}$  e  $\omega_{min}$  descrevem as frequências naturais máxima e mínima, respectivamente (DÖRFLER; BULLO, 2011).

### 3.3 Teoria de Redes Complexas

A partir do modelo de Kuramoto, foi possível descrever com sucesso diversos casos reais de sincronização, tais como redes de junções de Josephson, conjunto de lasers, redes neurais e oscilações de densidades de carga (ACEBRÓN et al., 2005). Contudo, uma das particularidades destes cenários refere-se ao fato dos osciladores compartilharem o mesmo padrão de conexões, isto é, os elementos do sistema são organizados em uma topologia de rede completamente conectada. Apesar do êxito na descrição destes cenários, uma dificuldade encontrada em modelos globalmente conectados refere-se ao aumento exponencial da complexidade do problema em função do número de elementos da rede.

Desta maneira, uma das principais limitações do modelo original de Kuramoto é, de fato, a aplicabilidade em redes complexas, cujos elementos da rede não dispõem de um acoplamento global. Uma rede pode ser descrita como uma coleção de elementos e conexões, também chamados de vértices e arestas. Por sua vez, redes complexas refere-se a um sistema composto por diversos elementos que interagem entre si, cujo comportamento coletivo não pode ser explicado por meio do comportamento individual de cada elemento.

#### 3.3.1 Definição de Redes Complexas

Em sua essência, a teoria de redes complexas objetiva compreender e estabelecer políticas para conexão entre pares de elementos agrupados sobre um determinado contexto (WASSERMAN; FAUST, 1994). Em outras palavras, a teoria de redes complexas visa estudar a natureza das ligações entre os elementos de uma rede, concentrando o foco sobre a análise das estruturas de ligação entre os elementos de uma rede (BARABÁSI, 2002).

A teoria de redes complexas possui um caráter interdisciplinar e isto têm instigado inúmeros cientistas de diferentes campos de pesquisa. Por exemplo, os conceitos de redes complexas e teoria de grafos são alicerces para o estudo de redes sociais em sociologia. Neste sentido, a topologia de uma rede social é definida pela maneira com a qual as

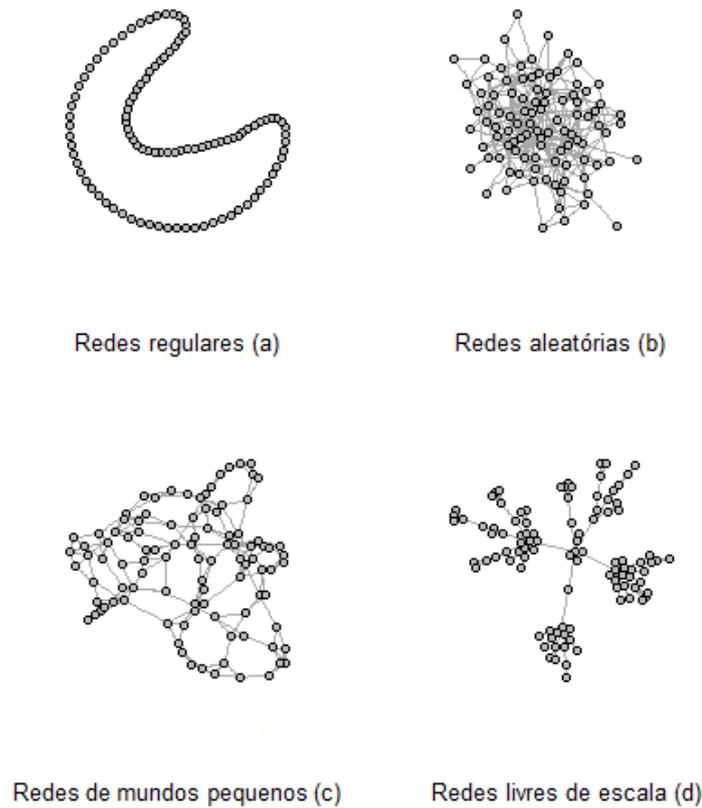


Figura 3.2: Exemplo de redes complexas com diferentes topologias. Gerado pelo software de computação estatística R (R Development Core Team, 2008).

ligações se formam dentro da rede. Este exemplo de rede complexa possui um caráter dinâmico, isto é, suas propriedades evoluem ao longo do tempo, seja por meio da adição ou remoção de novos elementos ou ligações. Dentre os principais modelos de redes complexas formulados, destacam-se as redes regulares, aleatórias, mundos pequenos e livre de escala (Figura 3.2). A seguir são descritas as propriedades destes principais modelos de redes.

### 3.3.2 Redes Regulares

O modelo de redes regulares é originário da teoria dos grafos, o qual é caracterizado pela homogeneidade das ligações, isto é, todos os elementos apresentam exatamente a mesma quantidade de ligações. Portanto, os elementos em uma rede regular devem possuir exatamente a mesma distribuição de grau, a qual representa a quantidade de arestas incidentes para cada vértice em um dado grafo. O modelo de redes regulares é usualmente utilizado para descrever a topologia de interação em sociedades artificiais, cujos elementos possuem uma localização geométrica e suas interações são limitadas aos vizinhos mais próximos (WASSERMAN; FAUST, 1994).

Formalmente, uma rede pode ser representada por um grafo  $G(\mathcal{V}, \mathcal{E})$ , composto por vértices  $\mathcal{V} = \{1, \dots, N\}$  e arestas  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . Em redes regulares, todos os vértices possuem o mesmo grau  $\deg(v)$ , ou seja, a mesma quantidade de arestas incidentes. Um grafo não direcionado é designado como completo quando todos os vértices possuem um grau  $\deg(v) = |\mathcal{V}| - 1$ . Apesar de pouco prático em aplicações reais, o modelo de redes regulares foi amplamente utilizado como base para descrever outros modelos mais elaborados de redes complexas (WASSERMAN; FAUST, 1994).

### 3.3.3 Redes Aleatórias

Por sua vez, o modelo de redes aleatórias, proposto por Erdős e Rényi, sugere a ideia de grafos desordenados, cujas ligações dos elementos da rede ocorrem de maneira aleatória (ERDÖS; RÉNYI, 1959). O modelo de redes aleatórias parte do princípio que a probabilidade de existir uma ligação entre um par de elementos é igual ou similar para todos os demais pares de elementos. Como principal característica, a distribuição das ligações entre os elementos de uma rede aleatória pode ser considerada altamente homogênea (FORTUNATO; CASTELLANO, 2012). Em virtude desta característica de distribuição das ligações entre os elementos, as redes aleatórias também são conhecidas como redes igualitárias.

As redes aleatórias podem ser descritas por um grafo  $G(\mathcal{V}, p)$  ou  $G(\mathcal{V}, m)$ , onde  $m$  representa o número de arestas e  $p$  representa a probabilidade de conexão entre cada par de vértices  $\{i, j\}$  da rede. Erdős e Rényi também introduziram o conceito de *cluster*, que expressa grupos de elementos conectados entre si (ERDÖS; RÉNYI, 1959). Considerando que as ligações em uma rede aleatória são, em sua essência, homogêneas, é bastante comum a formação de *cluster* dentro da rede. Para tal, o modelo de redes aleatórias requer no mínimo uma ligação por elemento para manter a rede conectada.

### 3.3.4 Redes de Mundos Pequenos

O trabalho pioneiro de Erdős e Rényi sobre redes aleatórias desencadeou uma série de novos estudos relacionados à dinâmica de formação de redes complexas. O pesquisador Stanley Milgram realizou uma análise experimental para observar o grau de separação entre os elementos em uma dada rede (MILGRAM, 1967). A partir de seu experimento, foi constatado que todo elemento possui uma pequena distância entre os demais na rede, evidenciando que todos vivem em um mundo pequeno (WATTS; STROGATZ, 1998).

Após Stanley Milgram publicar seu trabalho sobre o princípio dos seis graus de separação, diversos sociólogos começaram a estudar as propriedades das ligações e suas

implicações na rede. Entretanto, estas pesquisas apresentavam maior ênfase aos laços fortes de uma rede. Em geral, os laços fortes representam as ligações entre os indivíduos dentro de um determinado *cluster*. Assim, o sociólogo Mark Granovetter propôs uma nova perspectiva, onde os laços fracos representam as ligações mais importantes de uma rede (GRANOVETTER, 1973). Os laços fracos correspondem às ligações entre elementos que permitem conectar dois *clusters* distintos.

Uma rede de mundos pequenos é representada por um grafo  $G(\mathcal{V}, \mathcal{E})$  com uma dimensão inicial  $d$  e uma probabilidade  $p$  de reconexões, considerando cada par de vértices  $\{i, j\} \in \mathcal{E}$ . Como consequência, esta probabilidade de reconexões subsequentes produzirá novos *clusters* na rede. Desta forma, quando a probabilidade  $p \approx 0$ , a rede apresenta uma topologia similar à uma rede regular. Por outro lado, quando a probabilidade  $p \approx 1$ , a rede apresenta uma topologia próxima à uma rede aleatória.

### 3.3.5 Redes Livre de Escala

De fato, as ligações entre os elementos são descritas de forma aleatória tanto nos modelos de redes aleatórias quanto no modelo de mundos pequenos. Não obstante, ao se tratar de cenários reais, é improvável que a origem das ligações entre os vértices de uma rede complexa seja sobretudo aleatória. Adicionalmente, também é improvável que uma rede complexa seja igualitária, isto é, todos os elementos apresentem aproximadamente a mesma quantidade de ligações (BARABÁSI; ALBERT, 1999).

Estas questões permaneceram em aberto por muito tempo, até os pesquisadores Réka Albert e Albert-László Barabási discordarem da concepção de que as redes complexas são essencialmente aleatórias (BARABÁSI; ALBERT, 1999). Para estes pesquisadores, as redes complexas reais apresentam alguma ordem em sua formação e evolução, ao invés de simplesmente seguir uma escala previamente definida.

Neste sentido, Albert e Barabási observaram que grande parte das redes complexas reais possui um pequeno número de elementos altamente conectados. Estes elementos são denominados conectores ou *hubs* e geralmente possuem uma grande influência dentro da rede (BARABÁSI; ALBERT, 1999). Em contrapartida, os modelos aleatórios não suportam o conceito de *hubs*, visto que a ideia de elementos altamente conectados contraria a concepção de redes igualitárias (ALBERT; BARABÁSI, 2002).

Outro aspecto relevante observado por Albert e Barabási foi que a distribuição de graus dos elementos segue uma lei de potência (BARABÁSI; ALBERT, 1999). Isto é, as conexões de uma rede  $G(\mathcal{V}, \mathcal{E})$  evoluem seguindo uma função exponencial  $\gamma$  de conexões preferenciais. Portanto, da mesma forma que há poucos elementos altamente conectados

na rede, também há muitos elementos com poucas ligações.

Em outras palavras, como as redes complexas normalmente apresentam um caráter dinâmico, os elementos mais recentes da rede tendem a apresentar menos ligações do que os elementos mais antigos. A partir da constatação, Albert e Barabási afirmaram que redes complexas reais são livres de escalas devido à existência de *hubs* que fundamentalmente definem a topologia da rede e dominam o comportamento de novas conexões.

## 3.4 Sincronização em Redes Complexas

Diversos sistemas complexos reais apresentam padrões de conexão distintos entre os elementos da rede (DÖRFLER; M.; F., 2013). Processos de reações químicas, metabolismo celular, malhas de distribuição elétrica e clivagem proteolítica são alguns exemplos de sistemas complexos reais que apresentam diferentes topologias de rede e dinâmicas de conexão entre os elementos (BARABÁSI, 2002).

Além da distribuição de grau não uniforme entre os elementos da rede, muitos cenários reais também não dispõem de uma força do acoplamento homogênea. Alguns exemplos reais que requerem diferentes pesos entre as conexões incluem sistemas sociais, biológicos, transportes e econômicos (YOOK; JEONG; BARABÁSI, 2001). Deste modo, são necessárias algumas adaptações no modelo de Kuramoto tradicional para lidar com cenários reais cujas topologias de rede não são triviais (ARENAS et al., 2008).

### 3.4.1 Representação de Redes Complexas

A dinâmica da interação entre um conjunto de  $N$  osciladores de fase é tipicamente descrita através de um grafo não direcionado  $G(\mathcal{V}, \mathcal{E}, \mathcal{A})$ , composto por um conjunto de vértices  $\mathcal{V} = \{1, \dots, N\}$  e um conjunto de arestas  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ , cujos vértices representam osciladores e as arestas denotam o acoplamento entre pares de osciladores. As arestas  $\{i, j\} \in \mathcal{E}$  de um grafo são usualmente representadas por meio de uma matriz de adjacência  $a_{ij}$ , onde os pesos contidos em  $\mathcal{A}$  para cada aresta descrevem a força de acoplamento entre os osciladores  $i$  e  $j$ , representado por  $a_{ij} > 0$ .

Em uma rede de osciladores homogêneos globalmente acoplados, todas as arestas possuem o mesmo peso, onde  $\forall \{i, j\} \in \mathcal{E} \mid a_{ij} = 1$ . Em outras palavras,  $a_{ij} = \frac{K}{N} > 0$  reproduz o mesmo comportamento de sincronização original do modelo de Kuramoto. Por sua vez, em redes complexas os elementos não dispõem de um único padrão de conexão ou acoplamento homogêneo, portanto,  $\exists \{i, j\} \in \mathcal{E} \mid a_{ij} \neq 1$ . Assim, além da possibilidade dos osciladores apresentarem diferentes graus, a força de acoplamento também pode variar

entre pares  $\{i, j\}$  de osciladores acoplados.

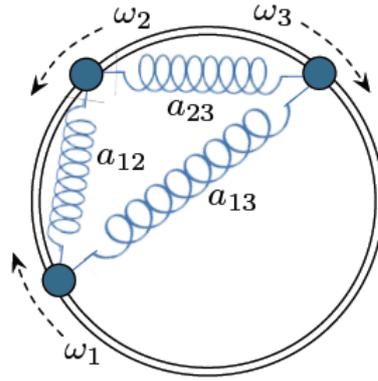


Figura 3.3: Exemplo de uma rede de osciladores globalmente acoplados com diferentes pesos em cada aresta. Adaptada de (DÖRFLER; M.; F., 2013).

A Figura 3.3 apresenta uma rede de osciladores globalmente acoplados, onde os osciladores deslocam-se sobre uma mesma órbita em frequências  $\omega$  diferentes, sem colidirem (DÖRFLER; M.; F., 2013). Pares  $\{i, j\}$  de osciladores acoplados estão conectados por molas elásticas com diferentes rigidezes, representadas por  $a_{ij}$ . Portanto, o movimento de cada oscilador  $i$  será influenciado pelas conexões  $a_{ij}$  com os demais osciladores, desde que estas molas elásticas possuam força suficiente para interferir na frequência inicial  $\omega_i$ .

Ao assumir que as arestas  $a_{ij}$  de uma determinada rede complexa possuem pesos diferentes, isto é, os osciladores não dispõem de um acoplamento global, é necessário incorporar ao modelo de Kuramoto a força do acoplamento específica para cada par de osciladores acoplados (ARENAS et al., 2008), conforme

$$\frac{d\theta_i}{dt} = \omega_i + \sum_{j=1}^N a_{ij} \sin(\theta_j - \theta_i) \quad (3.6)$$

onde  $a_{ij}$  representa a força de acoplamento  $\forall i \neq j$ . Assim, o modelo original de Kuramoto para redes de osciladores globalmente conectados é retomado por  $a_{i,j} = \frac{K}{N} \mid \forall \{i, j\} \in \mathcal{E}$  (ARENAS et al., 2008).

Determinar com precisão a condição de sincronização de fase em uma rede com padrões complexos de interação entre os osciladores tem sido objeto de estudo de diversos pesquisadores na atualidade. O trabalho de Dörfler et al. (DÖRFLER; M.; F., 2013) sugeriu uma condição de sincronização que descreve uma solução única e estável  $\theta^*$  com frequências sincronizadas e fases coesas  $|\theta_i^* - \theta_j^*| \leq \gamma \leq \frac{\pi}{2}$ , para cada aresta  $\{i, j\} \in \mathcal{E}$  se  $\|L^\dagger \omega\| \in \mathcal{E} \leq \sin(\gamma)$ , onde  $L^\dagger$  representa a matriz pseudoinversa.

Assim, a condição de sincronização sugere a existência de uma solução  $\theta^*$  única

e estável considerando o acoplamento crítico  $K_c \leq \frac{\pi}{2}$  se  $\|L^\dagger \omega\|_{\mathcal{E}, \infty} < 1$ . Esta condição foi demonstrada através de avaliações numéricas do acoplamento crítico  $K_c$  exato para a ocorrência de sincronização em redes complexas, considerando modelos de redes regulares, redes aleatórias e redes de mundos pequenos (DÖRFLER; M.; F., 2013).

### 3.4.2 Redes Complexas Adaptativas

Em geral, o modelo de Kuramoto e suas variações presumem que o acoplamento entre osciladores seja constante, isto é, o acoplamento permanece imutável à medida que o processo de sincronização evolui ao longo do tempo. Desta forma, a topologia da rede mantém-se estática enquanto os estados dos osciladores alteram dinamicamente. Tais modelos matemáticos visam, em sua essência, investigar o comportamento coletivo produzido pela dinâmica individual dos osciladores na rede.

Contudo, redes complexas reais normalmente apresentam propriedades evolutivas que alteram a topologia ou até mesmo a força de acoplamento entre os elementos da rede. Para lidar com estes cenários, modelos de redes complexas adaptativas têm sido explorados recentemente em diferentes campos de pesquisa (GROSS; BLASIUS, 2008). Sobre o contexto de sistemas dinâmicos e sincronização de fase, estudos bem-sucedidos foram realizados sobre reforço dos acoplamentos e reconexões entre pares de osciladores (ITO; KANEKO, 2001; ZHOU; KURTHS, 2006; TANAKA; AOYAGI, 2008).

Especificamente sobre reforço dos acoplamentos, a técnica consiste em recalibrar os pesos  $a_{ij}$  entre as arestas ao longo do tempo afim de otimizar o processo de sincronização da rede. Esta técnica pode acelerar o processo de sincronização ou até mesmo forçar a convergência do processo em relação à força de acoplamento, tornando a rede passível de sincronização (ZHOU; KURTHS, 2006).

Em termos gerais, a evolução da topologia em uma rede adaptativa depende da dinâmica local dos osciladores. Deste modo, o processo de evolução da rede ocorre de forma iterativa, sendo a topologia da rede influenciada pelo comportamento individual dos osciladores ao longo do tempo. Por conseguinte, o comportamento individual dos osciladores também evolui em resposta às mudanças na topologia, como ilustrado pela Figura 3.4. Portanto, tais modelos adaptativos possibilitam compreender o impacto da topologia sobre a dinâmica individual de sincronização dos elementos em redes complexas que evoluem ao longo do tempo.

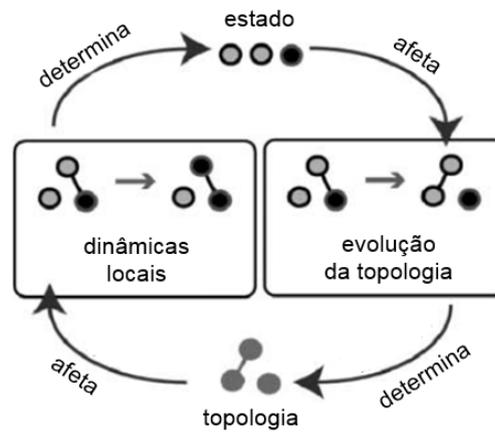


Figura 3.4: Processo iterativo de evolução em redes complexas adaptativas. Adaptada de (GROSS; BLASIUS, 2008).

### 3.4.3 Estabilidade em Redes Complexas Sincronizadas

Outra importante linha de pesquisa sobre sincronização em redes complexas refere-se à investigação da estabilidade do estado síncrono entre os osciladores acoplados do sistema (ARENAS et al., 2008). Esta linha de pesquisa foi introduzida por Pecora, Carroll e Barahona (PECORA; CARROLL, 1998; BARAHONA; PECORA, 2002) ao investigar a estabilidade da sincronização em uma rede através da Função de Estabilidade Mestre, ou *Master Stability Function* (MSF). O MSF possibilita calcular a estabilidade de um estado do sistema ou um estado particular de um oscilador, considerando medidas de estabilidade como o expoente de Lyapunov (ARENAS et al., 2008).

Neste sentido, o MSF permite mapear períodos de instabilidade da rede, bem como o efeito da adição de ruídos ou interferências externas sobre o estado síncrono da rede (PECORA; CARROLL, 1998). No entanto, para esta formulação é necessário assumir que os osciladores da rede são idênticos, representados por um vetor de estados em um espaço  $m$ -dimensional na forma  $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ .

Além disto, os osciladores devem compartilhar uma única função de acoplamento. Esta função de acoplamento, caracterizada por  $\mathbf{H}(\mathbf{x})$ , produz um sinal em resposta ao estado  $\mathbf{x}$ , onde este sinal é propagado aos demais osciladores acoplados. Portanto, a função  $\mathbf{H} : R^m \rightarrow R^m$  representa o impacto do acoplamento através de uma matriz de dimensão  $m \times m$  (PECORA; CARROLL, 1998). Para uma rede de  $N$  osciladores acoplados representada por uma matriz de adjacência  $a_{ij}$ , a formulação MSF é descrita por

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}_i) + \sigma \sum_{j=1}^N a_{ij} \mathbf{H}(\mathbf{x}_j) \quad (3.7)$$

onde  $a_{ij}$  representa a topologia da rede e pesos das arestas  $\forall i \neq j$  e  $\sigma$  refere-se à força do acoplamento global da rede (PECORA; CARROLL, 1998).

Nestas condições, o estado síncrono do sistema é alcançado no momento em que todos os osciladores compartilharem um único estado, conforme

$$\mathbf{x}_1(t) = \mathbf{x}_2(t) = \dots = \mathbf{x}_N(t) = s(t) \quad (3.8)$$

onde  $s(t)$  representa a solução para a Equação 3.7. Em outras palavras,  $s(t)$  descreve o subespaço em que todos os osciladores evoluem de forma síncrona, cujo comportamento é conhecido como sincronização mútua (ARENAS et al., 2008).

Portanto, a sincronização mútua descrita pela Equação 3.8 é alcançada quando a função  $\mathbf{H}(\mathbf{x})$  for igual para todo e qualquer oscilador  $i$ . Desta maneira, a estabilidade da sincronização é garantida enquanto o comportamento do sistema permanecer dentro do espaço de sincronização mútua. Um dos principais objetivos desta análise corresponde à investigação da influência de eventuais perturbações externas sobre o sistema que podem comprometer o estado de sincronização mútua (ARENAS et al., 2008).

#### 3.4.4 Expoente de Lyapunov

Em sua essência, o expoente de Lyapunov possibilita medir a sensibilidade do comportamento de sincronização da rede em relação aos parâmetros iniciais do sistema. Mais especificamente, através do expoente de Lyapunov é possível descrever a frequência em que dois pontos se aproximam ou afastam em trajetórias distintas dentro de um espaço de fases de um sistema (PECORA et al., 1997).

O expoente de Lyapunov é caracterizado como a taxa de separação entre trajetórias distintas, dado por

$$\lambda = \lim_{x \rightarrow \infty} \lim_{\delta x_0 \rightarrow \infty} \frac{1}{t} \ln \frac{|\delta x(t)|}{|\delta x_0|} \quad (3.9)$$

onde  $\delta x_0$  representa a diferença inicial entre duas órbitas e  $\delta x(t)$  indica a diferença entre duas órbitas em um intervalo de tempo  $t$  (CENCINI; CECCONI; VULPIANI, 2010). Por conseguinte, a dinâmica em um espaço de fases  $m$ -dimensional de um sistema forma o espectro de Lyapunov descrito pela série  $\lambda_1, \lambda_2, \dots, \lambda_m$  (PECORA et al., 1997).

De maneira geral, o expoente de Lyapunov é utilizado para avaliar a estabilidade do estado síncrono de um sistema considerando pequenas perturbações em sua trajetória (CENCINI; CECCONI; VULPIANI, 2010). Esta avaliação é realizada tomando como base o expoente máximo de Lyapunov  $\lambda_{max}$  de um espaço de fases para um dado sistema. Caso  $\lambda_{max} < 0$ , o sistema mantém-se estável visto que todos os demais expoentes de

Lyapunov são menores que  $\lambda_{max}$ . Caso contrário, o sistema é instável quando  $\lambda_{max} \geq 0$ , indicando que um dos expoentes de Lyapunov também é positivo e o sistema apresenta uma divergência de trajetórias (PECORA et al., 1997).

## 3.5 Aplicações da Sincronização

O modelo de Kuramoto fornece uma maneira simples e natural para descrever o comportamento coletivo em sistemas dinâmicos reais. Neste sentido, a simplicidade do modelo de Kuramoto, aliado ao caráter multidisciplinar, têm estimulado sua aplicação em diferentes áreas de pesquisa com características comuns. Estudos sobre dinâmicas de sincronização e redes complexas, por exemplo, são temas muito explorados na ciência da computação, sobretudo ao se tratar de sistemas distribuídos (ARENAS et al., 2008). A seguir são apresentados alguns trabalhos notáveis referentes à aplicação de conceitos da sincronização física, mais especificamente o modelo de Kuramoto, para lidar com alguns desafios presentes em sistemas distribuídos.

### 3.5.1 Sistemas Distribuídos

A sincronização em sistemas distribuídos é uma característica recomendável para obter um desempenho eficiente nas interações entre as unidades de software autônomas. Um dos principais objetivos de um sistema distribuído é alcançar um estado global comum entre estas unidades de software que, em geral, possuem interesses locais distintos ou competem pelos mesmos recursos (FISCHER, 1983).

As maiores dificuldades na resolução de problemas desta natureza frequentemente estão relacionadas a questões envolvendo escalabilidade e a complexidade da topologia do sistema distribuído, principalmente quando o ambiente se refere à uma rede complexa em larga escala (SIVRIKAYA; YENER, 2004). Nos últimos anos foram realizadas diversas pesquisas com sucesso aplicando modelos de sincronização física em redes de osciladores acoplados em cenários clássicos de sistemas distribuídos, tais como smart-grids, para controle distribuído de densidades de carga (DORFLER; BULLO, 2012).

### 3.5.2 Controle de Densidades de Carga e Smart-Grids

Redes de energia são exemplos reais de redes complexas de mundos pequenos, em geral constituídas por inúmeros *clusters* de estações de consumo e de geração de energia conectadas por linhas de distribuição (WATTS; STROGATZ, 1998). Estas redes

de energia são auto-organizáveis, pois crescem e alteram suas ligações sem a necessidade de um controle central. Neste sentido, as redes de energia devem apresentar mecanismos de tolerância à falhas e estratégias para maximizar a eficiência energética ao longo da malha de distribuição (BARABÁSI, 2002).

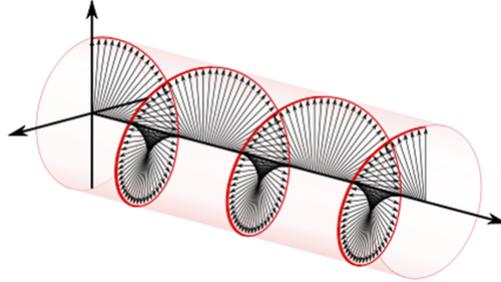


Figura 3.5: Sincronização do fluxo de densidade de carga em uma rede de osciladores.

Um dos desafios comuns neste tipo de aplicação refere-se à sincronização de um número grande de estações afim de estabilizar e aumentar a energia efetiva de saída. Este cenário também foi observado e bastante explorado em redes de Junções de Josephson e conjuntos de lasers (ACEBRÓN et al., 2005). As redes de Junções de Josephson, por exemplo, são sistemas formados por supercondutores que oscilam em altas frequências, cujo propósito da sincronização consiste em regular as frequências dos dispositivos para potencializar o desempenho do sistema (Figura 3.5).

Uma rede elétrica com corrente alternada pode ser descrita por um grafo ponderado não dirigido, formado por vértices  $\mathcal{V} = \{1, \dots, N\}$  e linhas de transmissão  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . Neste sentido, o modelo de Kuramoto foi ligeiramente adaptado para lidar com as especificidades deste cenário, conforme

$$M_i \ddot{\theta}_i = \omega_i + \sum_{j=1}^N a_{ij} \sin(\theta_j - \theta_i) \quad (3.10)$$

onde  $M_i$  representa o coeficiente de inércia e  $\ddot{\theta}_i$  refere-se ao coeficiente de amortecimento viscoelástico do sistema (DÖRFLER; BULLO, 2012). Desta maneira, o comportamento coletivo dos osciladores irá conduzir à sincronização das fases por meio dos acoplamentos  $a_{ij} \sin(\theta_j - \theta_i)$  entre os osciladores vizinhos.

Caso todas as frequências naturais  $\omega_i$  forem idênticas, o processo de sincronização produzirá um equilíbrio de fase no sistema, isto é, os ângulos  $\theta_i$  serão alinhados. Já em situações onde as frequências naturais  $\omega_i$  são heterogêneas, a tendência do sistema manterá as oscilações fora de fase quando não houver um acoplamento adequadamente distribuído entre os pesos das arestas. Por sua vez, o acoplamento crítico é substancialmente sensível à topologia e aos parâmetros iniciais do sistema (DÖRFLER; M.; F., 2013).

### 3.5.3 Comunicação em Redes Wireless

Redes wireless *ad-hoc* correspondem à redes dinâmicas, auto-organizáveis e auto-configuráveis, cujos nós wireless conectam-se e formam uma rede sem a necessidade de uma infraestrutura predefinida. Em função desta ausência de infraestrutura, redes *ad-hoc* não dispõem de dispositivos centralizadores, tais como roteadores ou pontos de acesso centralizados. Desta maneira, em uma rede wireless *ad-hoc* os nós devem cooperar para enviar os pacotes de dados da origem até o destino através de uma técnica conhecida como *multi-hop* (SIVRIKAYA; YENER, 2004).

Assim como em outras aplicações envolvendo sistemas distribuídos, a sincronização temporal é necessária para alinhar os dados transmitidos entre os diversos dispositivos wireless que compõem a rede (SIVRIKAYA; YENER, 2004). Isto é, a sincronização temporal provê uma única escala de tempo de referência para os relógios locais de cada dispositivo wireless. O desalinhamento temporal dos relógios ocorre sobretudo devido à falta de precisão dos equipamentos e também pelo fato dos dispositivos wireless não serem ativados simultaneamente (SIMEONE; SPAGNOLINI, 2007). Além disto, devido a característica dinâmica e auto-organizável de redes wireless *ad-hoc*, este cenário deve considerar também a adição e remoção de dispositivos ao longo do tempo.

Diante disto, diversos algoritmos para sincronização temporal em redes wireless têm sido propostos até então. Contudo, os algoritmos dependem em algum momento da troca de mensagens entre os dispositivos wireless. Os desafios deste cenário envolvem desde o custo de propagação das mensagens de atualização até o consumo de energia eficiente dos dispositivos em casos mais restritivos como em redes de sensores. O trabalho de Pagliari e Scaglione (PAGLIARI; SCAGLIONE, 2011) demonstrou experimentalmente a eficiência do modelo de sincronização em redes de osciladores acoplados aplicado sobre tal cenário de alinhamento temporal em redes wireless.

Uma rede wireless pode ser representada por meio de um grafo não dirigido, onde a topologia da rede normalmente segue um modelo geométrico e aleatório. Díaz-guilera et al. (DÍAZ-GUILERA et al., 2009) realizaram diversos experimentos em redes com esta característica afim de investigar as condições de sincronização através do modelo de Kuramoto. Os experimentos realizados também consideraram novas conexões entre os dispositivos wireless com uma probabilidade  $p$ , além do aumento gradual da força de acoplamento  $K$  entre intervalos de tempo.

A Figura 3.6 ilustra a dinâmica de sincronização temporal através de uma rede de osciladores acoplados. Neste exemplo, um oscilador (b) ajusta gradualmente as fases de acordo com os períodos do outro oscilador (a) para cada tempo  $t$ , através de atrasos

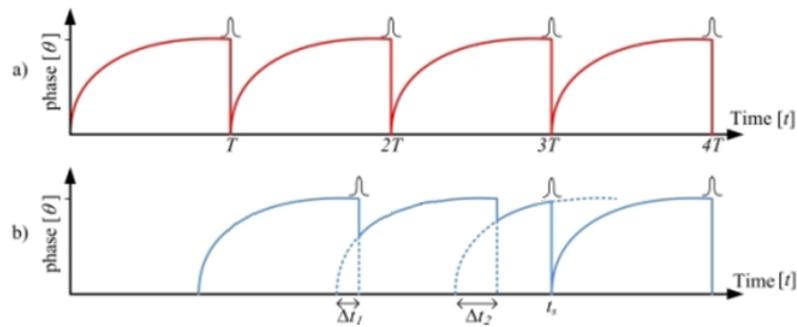


Figura 3.6: Dinâmica de sincronização temporal em uma rede de osciladores acoplados.

temporais na frequência do oscilador. Outra aplicação comum do modelo de sincronização de Kuramoto sobre o contexto de redes wireless refere-se ao compartilhamento de canais de frequência em redes TDMA (*Time Division Multiple Access*). Os pesquisadores Degesys et al. (DEGESYS et al., 2007) introduziram um novo algoritmo de escalonamento de janelas de tempo de canais inspirado no comportamento de osciladores acoplados, comparando o desempenho deste método proposto com outras abordagens clássicas de escalonamento em redes TDMA.

### 3.5.4 Protocolos de Consenso

Em sistemas distribuídos, o consenso representa a habilidade de um conjunto de entidades autônomas alcançarem um valor único e comum em um estado estacionário e assintótico. Em outras palavras, tais entidades autônomas, denominadas agentes de software, devem consentir sobre seus estados mesmo que apresentem interesses individuais diferentes. Portanto, o aspecto chave em problemas de consenso não é a investigação dos interesses individuais dos agentes, e sim o fato de como devem cooperar para alcançarem a mesma conclusão ou resultado (FISCHER, 1983).

A sincronização de fase em rede de osciladores acoplados pode ser descrita como um problema de consenso, onde o objetivo é alcançado por meio da sincronização da frequência dos osciladores (ARENAS et al., 2008). Nesta analogia, a sincronização representa o consenso e os osciladores referem-se aos agentes do problema. Como consequência, o consenso é alcançado pela influência dos estímulos recebidos de cada agente vizinho.

Problemas de consenso são usualmente representados por meio de grafos, onde os vértices representam os agentes e as arestas correspondem às restrições. Em outras palavras, a topologia do grafo descreve o padrão complexo de interações entre os agentes. A dinâmica de cada agente evolui ao longo do tempo conforme  $\dot{x} = f(x_i, u_i)$ , onde  $x_i$  representa um possível valor para o estado do agente  $u_i$  e a função descreve a influência

deste estado perante aos vizinhos. Portanto, dois agentes conectados  $u_i$  e  $u_j$  alcançam o consenso se e somente se  $x_i = x_j$  (OLFATI-SABER; MURRAY, 2004).

Mais especificamente, dado  $\chi : \mathfrak{R}^n \rightarrow \mathfrak{R}$  como a função de  $n$  variáveis  $x_1, \dots, x_n$  e  $a = x(0)$  o estado inicial do sistema, o equilíbrio  $x^*$  em um protocolo de consenso  $\chi$  é obtido  $x_i^* = \chi(x(0))$ . Um modelo de referência para protocolo de consenso pode ser representado por (ARENAS et al., 2008)

$$\dot{x}_i = \sum_{j=1}^N a_{i,j}(t)(x_j(t) - x_i(t)) \quad (3.11)$$

cujo comportamento assintótico do sistema é análogo à formulação MSF para análise de estabilidade da rede (ver Sessão 3.4.3). Assim, os pesquisadores Olfati-Saber e Murray (OLFATI-SABER; MURRAY, 2004) estudaram modelos de sincronização física aplicados sobre problemas de consenso, considerando aspectos como atraso temporal e mudanças na topologia da rede.

### 3.5.5 Movimento e Controle Coletivo

Controle coletivo é um importante e crescente campo de pesquisa em sistemas distribuídos, o qual está relacionado às aplicações que requerem a coordenação de grupos de sistemas autônomos (PALEY et al., 2007). Exemplos de aplicações neste contexto incluem veículos não tripulados, controle de tráfego aéreo, redes de sensores e sistemas de rastreamento ou monitoramento. Especificamente sobre movimento coletivo, a formação e o alinhamento das trajetórias dos agentes são alguns desafios que compõem o estudo sobre controle distribuído (SEPULCHRE; PALEY; LEONARD, 2008).

Os modelos de movimento coletivo normalmente seguem critérios como separação física mínima entre os agentes, alinhamento e coesão do deslocamento ao longo do tempo (SEPULCHRE; PALEY; LEONARD, 2008). Tal controle de formação e rastreamento de trajetória objetiva a convergência do movimento de um conjunto de agentes móveis, evitando possíveis colisões. Além disto, há também a necessidade dos agentes possuírem líderes virtuais que definem a trajetória a ser seguida, cujo modelo de comunicação e troca de informações requeridas pelo controle coletivo dos agentes são descritas por meio de grafos (KLEIN; LALISH; MORGANSEN, 2009).

Neste sentido, os pesquisadores Paley et al. (PALEY et al., 2007) abordaram o cenário de formação e rastreamento de trajetórias em sistemas baseados em agente de maneira análoga à sincronização em redes de osciladores acoplados. Já Klein, Lalish e Morgansen (KLEIN; LALISH; MORGANSEN, 2009) investigaram o uso de modelos de

sincronização física em uma abordagem típica para movimento coletivo, conhecida como líder-seguidor. Nesta abordagem, os métodos de controle são realimentados e influenciados por um ou mais líderes em um grupo de agentes.

## 3.6 Considerações Finais

A sincronização física em redes de osciladores acoplados é um campo de pesquisa em expansão, a qual possibilita a representação de inúmeros problemas complexos reais, em especial, ambientes em larga escala. O modelo de Kuramoto tem se demonstrado uma formulação robusta o suficiente para representar uma grande variedade de padrões de sincronização. Adicionalmente, este modelo também é altamente flexível e adaptável para diferentes contextos de sincronização, incluindo cenários de consenso ou otimização (ARENAS et al., 2008). Como consequência, inúmeras variações do modelo de Kuramoto têm sido investigadas na literatura.

Neste sentido, os conceitos da sincronização física em redes de osciladores, bem como o modelo de Kuramoto, são fontes de inspiração promissoras para abordar os grandes desafios inerentes à coordenação descentralizada em sistemas multiagente, tais como a capacidade de convergência, equilíbrio e consenso dos agentes. Mais especificamente, tais princípios da sincronização física podem contribuir para uma nova perspectiva de pesquisa sobre problemas de otimização distribuída de restrição, visando oferecer uma maior robustez e escalabilidade em relação aos métodos atuais para resolução de DCOP.

## Capítulo 4

# Otimização Distribuída de Restrição Baseada em Sincronização Física

A investigação sobre modelos sincronização física em redes de osciladores acoplados tem despertado o interesse de inúmeros pesquisadores, sobretudo, devido à capacidade de convergência, escalabilidade e aplicabilidade em vários sistemas dinâmicos reais. Um dos modelos amplamente adotado em tais estudos é caracterizado por uma população heterogênea de osciladores de fase, cujos acoplamentos existentes são representados por meio de grafos. Os estudos destes modelos visam compreender a dinâmica de interação dos elementos da rede, enfatizando aspectos como coerência e estabilidade do sistema.

Este capítulo tem como objetivo introduzir um novo método para resolução de problemas de otimização distribuída de restrição baseado em sincronização física, o qual constitui o propósito geral deste trabalho. Mais especificamente, este método proposto, denominado *Coupled Oscillators OPTimization* (COOPT), captura a essência do modelo de sincronização física de Kuramoto e a aplica sobre o processo de resolução de DCOP. Deste modo, o presente capítulo aborda em detalhes a estratégia do COOPT e como este método para coordenação descentralizada em sistemas multiagente é capaz de acelerar o processo de convergência das soluções candidatas, além de oferecer maior escalabilidade em ambientes complexos em larga escala.

### 4.1 Correlação entre DCOP e Redes de Osciladores

De maneira geral, um DCOP apresenta aspectos muito similares ao processo de sincronização física em uma rede de osciladores acoplados. Em resumo, um DCOP objetiva a coordenação das ações dos agentes a fim de otimizar a função objetivo global definida por um conjunto de restrições. Por outro lado, o processo de sincronização em uma rede de osciladores acoplados refere-se à convergência das frequências e à coesão das

fases entre os osciladores acoplados. Portanto, em ambos os casos, os objetivos estão relacionados à coerência global, ou consenso, entre grupos de indivíduos interagindo entre si em uma determinada rede.

#### 4.1.1 Variáveis como Osciladores

Uma analogia intuitiva entre a definição de um DCOP e o processo de sincronização em uma rede de osciladores acoplados consiste em representar as variáveis como osciladores e as restrições como acoplamentos entre os osciladores. Nesta analogia, um agente  $a_i$  detém um comportamento individual similar a um oscilador  $i$ , respondendo aos estímulos produzidos por seus vizinhos. Por sua vez, os respectivos valores  $d \in D_i$  compõem o domínio da variável  $x_i$ , onde  $d_i$  refere-se ao estado atual desta variável. Por conveniência, será assumido ao longo deste capítulo que cada agente  $a_i$  controla uma única variável  $x_i$  e os termos agente e variável podem ser utilizados de forma intercambiável.

**Definição 1.** Agente: cada agente  $a_i$  possui uma visão local sobre o problema, denominada *view*, a qual representa um conjunto parcial de atribuições das variáveis vizinhas definido por uma estrutura no formato  $\{(x_j, d_j), \dots\}$ . As informações contidas na visão local referem-se às atribuições das variáveis  $x_j \in N_i$ , onde  $N_i$  representa o conjunto de variáveis que compartilham alguma restrição com  $x_i$ .

**Definição 2.** Custos: o custo local  $\delta(d_i)$  para o valor  $d_i$  quando atribuído à variável  $x_i$  é definido por  $\delta(d_i) = \sum_{x_j \in view} f_{ij}(d_i, d_j)$ , quando  $x_i \leftarrow d_i$  e  $x_j \leftarrow d_j$ , de acordo com a visão local *view* do agente  $a_i$ . Por sua vez,  $\delta$  representa o custo global de uma determinada solução completa, ao passo que  $\delta^*$  representa o melhor custo global para um dado problema.

Portanto, o custo  $\delta(d_i)$  descreve o impacto local do valor  $d_i$  atribuído à variável  $x_i$ , ao considerar os demais valores das variáveis vizinhas. Em outras palavras, este impacto pode ser compreendido como a influência da ação  $x_i \leftarrow d_i$  sobre uma solução parcial do problema. Por sua vez, o modelo de sincronização física de Kuramoto descreve a frequência  $\omega_i$  do oscilador  $i$  como tal influência.

**Definição 3.** Frequência: os custos locais  $\delta(d)$  para cada valor  $d \in D_i$ , considerando as respectivas combinações de atribuição das variáveis vizinhas contidas em *view*, compõem o conjunto de possíveis frequências  $\omega_i$  para a variável  $x_i$ .

Entretanto, uma solução localmente ótima para um determinado agente pode apresentar conflitos de interesse entre os demais agentes envolvidos, devido aos objetivos

individuais dos agentes e suas visões parciais sobre o problema. Não obstante, o modelo de sincronização física de Kuramoto representa este conflito a partir da diferença das fases  $\theta$  entre os osciladores, conforme descrito pelo parâmetro de ordem  $r(t)$  utilizado para mensurar a coerência do sistema (Equação 3.5).

**Definição 4.** Fase: a fase  $\theta_i$  representa a discordância, isto é, um grau de erro, entre o custo local  $\delta(d_i)$  de uma variável  $x_i$  para uma dada solução parcial e respectivo custo global  $\delta$  para a solução completa, em um determinado momento.

#### 4.1.2 Acoplamentos e Funções de Custo

Uma das principais diferenças entre um DCOP e o modelo de Kuramoto reside no conceito de acoplamento. No modelo original de Kuramoto, a força de acoplamento é descrita de maneira constante e homogênea entre todos os osciladores acoplados da rede (KURAMOTO, 1984). Ou seja, no modelo de Kuramoto é assumido que todos os osciladores estão globalmente acoplados, formando uma topologia completa. No entanto, diversos estudos descrevem variações deste modelo original onde a força de acoplamento está dispersa em uma topologia com padrões complexos de interação entre os osciladores (ZHOU; KURTHS, 2006; GROSS; BLASIUS, 2008; DÖRFLER; M.; F., 2013).

Assim, os pesquisadores Dörfler e Bullo (DÖRFLER; M.; F., 2013) analisaram a dinâmica emergente em uma adaptação do modelo de Kuramoto para lidar com redes complexas de diferentes topologias e diferentes forças de acoplamento  $a_{ij}$  entre pares  $\{i, j\}$  de osciladores. Este estudo enfatizou o impacto desta adaptação do modelo de Kuramoto sobre o processo de sincronização dos osciladores, além de discutir sobre critérios para convergência da rede. De maneira geral, tal adaptação demonstrou uma estabilidade no processo de convergência, em especial, em redes regulares e em outros exemplos de redes cuja topologia apresenta uma distribuição de grau homogênea.

Em um DCOP, o acoplamento entre pares de variáveis  $x_i$  e  $x_j$  refere-se à uma restrição do problema, cuja representação é dada por meio de funções de custo  $f_{ij}$ . No entanto, tal acoplamento em um DCOP não é constante como no modelo original de Kuramoto, no sentido que os estados  $d_i$  e  $d_j$  das variáveis  $x_i$  e  $x_j$  impactam diretamente sobre o produto da função de custo  $f_{ij}(d_i, d_j)$ . Portanto, modelos de sincronização com pesos constantes  $a_{ij}$  não são favoráveis para aplicação em um DCOP.

Neste sentido, a investigação sobre modelos de sincronização física baseados em acoplamentos dependentes dos estados dos osciladores (SKARDAL; RESTREPO, 2012; MEIER; HASCHKE; RITTER, 2014; GUSHCHIN; MALLADA; TANG, 2015) surge como uma alternativa para o contexto de restrição em um DCOP. Nestes modelos, a força dos

acoplamentos é comumente representada por funções simétricas  $f(x_i, x_j) \equiv f(x_j, x_i)$  de interação entre os osciladores. Assim, o processo de atualização de fase do modelo de Kuramoto deve ser ligeiramente adaptado para considerar uma força de acoplamento dinâmica com base nos estados atuais dos osciladores, conforme

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^N f_{ij}(v_i, v_j) \sin(\theta_j - \theta_i) \quad (4.1)$$

onde  $v_i$  e  $v_j$  representam os estados atuais dos osciladores  $i$  e  $j$ .

Meier, Haschke e Ritter (MEIER; HASCHKE; RITTER, 2014) descreveram a força de acoplamento global  $K$  de maneira distribuída ao longo da rede por meio de funções simétricas para cada acoplamento entre pares  $\{i, j\}$  de osciladores. Este trabalho propôs uma nova abordagem para agrupamento perceptual baseado em redes de osciladores acoplados. Mais especificamente, o trabalho introduziu uma técnica de *clustering* para aprendizagem competitiva em redes neurais artificiais a partir de uma variação do modelo de Kuramoto para lidar com acoplamentos representados por funções simétricas.

### 4.1.3 Processo de Sincronização

A atualização gradual das frequências  $\omega_i$  é uma etapa fundamental no processo de sincronização em uma rede de osciladores. Quando há uma força de acoplamento suficiente, a dinâmica de sincronização conduz os osciladores à uma frequência média  $\Omega$  ao redefinir as fases de cada oscilador. Este processo objetiva reduzir a diferença existente entre as fases  $\theta_i$  de cada oscilador. Como visto anteriormente, para o contexto de um DCOP, a frequência é descrita como o custo local  $\delta(d)$  de uma variável, o qual refere-se ao impacto de uma ação local considerando os estados atuais das variáveis vizinhas.

Não obstante, modelos tradicionais de sincronização física consideram frequências contínuas, cujos estados dos osciladores são normalmente representados por intervalos circulares  $[0, 2\pi]$ . Por sua vez, em um DCOP os custos locais  $\delta(d)$  são discretos e finitos, ou seja, os custos são restritos às possíveis combinações de valores entre uma variável e seus vizinhos. Mais especificamente, a dinâmica de atualização das frequências deve considerar um conjunto discreto e finito  $d \in D_i$  de possíveis estados candidatos para um agente  $a_i$ , de acordo com a solução parcial contida em *view* em um dado momento.

Além disto, a atualização das frequências em um DCOP deve objetivar o consenso global das ações de cada agente, mesmo que existam ações localmente conflitantes. Neste sentido, pesquisas relacionadas à detecção de *clustering* em redes de osciladores acoplados (ARENAS; DÍAZ-GUILERA; PÉREZ-VICENTE, 2006; HONG; STROGATZ, 2011; WU

et al., 2011; MEIER; HASCHKE; RITTER, 2014) são opções candidatas para lidar com tais problemas de consenso, os quais naturalmente requerem a coordenação dos osciladores para formação de grupos.

No trabalho de Arenas, Díaz-Guilera e Pérez-Vicente (ARENAS; DÍAZ-GUILERA; PÉREZ-VICENTE, 2006), foi demonstrado que os modelos de sincronização baseados em campo médio são ineficientes para identificar os efeitos da dinâmica local dos osciladores. A partir desta observação, os pesquisadores introduziram um parâmetro de ordem local para mensurar a correlação dos estados entre pares  $\{i, j\}$  de osciladores, ao invés de considerar somente uma observação global, conforme

$$\rho_{ij}(t) = \langle \cos(\theta_j - \theta_i) \rangle \quad (4.2)$$

onde o cosseno da diferença das fases refere-se a uma medida de similaridade entre os osciladores vizinhos. Com isto é possível detectar a existência de grupos funcionais ou comunidades entre os osciladores em uma determinada rede.

Os pesquisadores Hong e Strogatz (HONG; STROGATZ, 2011) propuseram uma generalização do modelo de Kuramoto para distinguir os osciladores que são atraídos pelo campo médio e apresentam uma tendência de sincronização dos osciladores que são repelidos pelo campo médio e não possuem tendência de sincronização. Inspirados nos neurônios excitatórios e inibitórios observados em redes neurais, os autores substituíram a constante  $K$  por uma variável  $K_i$ . Assim, acoplamentos positivos  $K_i > 0$  representam neurônios excitatórios e os acoplamentos negativos  $K_i < 0$  descrevem neurônios inibitórios. Como consequência, esta formulação produz dois níveis de atração dos acoplamentos, separando os osciladores entre cooperativos e competitivos.

O trabalho de Wu et al. (WU et al., 2011) apresentou um método de *clustering* para resolução do problema de coloração de grafos. Este método é uma generalização do modelo de Kuramoto que também utiliza acoplamentos positivos e negativos, contudo, os acoplamentos negativos são aplicados quando um dado par  $\{i, j\}$  de osciladores pertence ao conjunto de arestas do grafo complementar à rede original. Além disto, o método proposto também utilizou a medida de similaridade baseada no cosseno para realizar o agrupamento dos osciladores, onde cada grupo formado representa uma cor no problema de coloração de grafos.

Os pesquisadores Meier, Haschke e Ritter (MEIER; HASCHKE; RITTER, 2014), a partir da técnica para aprendizagem competitiva já mencionada, também introduziram uma variação na dinâmica de atualização das frequências a fim de representar osciladores com características similares. Tais características, descritas por  $\alpha \in \mathcal{N}(\alpha)$ , encontram-se

distribuídas entre  $\mathcal{N}(\alpha) = \{1, \dots, L\}$  camadas de osciladores, cujas frequências evoluem iterativamente conforme

$$S_i(\alpha) = \sum_{i \in \mathcal{N}(\alpha)} f_{ij} \frac{1}{2} (\cos(\theta_j - \theta_i) + 1) \quad (4.3)$$

$$\omega_i = \omega_0 \operatorname{argmax}(S_i(\alpha)) \quad (4.4)$$

cuja função  $\operatorname{argmax}$  visa o agrupamento perceptual dos osciladores que compartilham características mais similares.

Assim, os trabalhos de (ARENAS; DÍAZ-GUILERA; PÉREZ-VICENTE, 2006) e (MEIER; HASCHKE; RITTER, 2014) foram utilizados como base para a elaboração de um modelo de coordenação para DCOP baseado em sincronização física. O processo de sincronização elaborado contempla um mecanismo de otimização que promove um equilíbrio entre a redução do custo individual e a redução do grau de discordância entre grupos de agentes. Desta maneira, o processo de sincronização evolui conforme

$$S_i = \sum_{x_j \in \text{view}} f_{ij}(d, d_j) + \frac{1}{2} f_{ij}(d, d_j) (\cos(\theta_j - \theta_i) + 1) \quad \forall d \in D_i \quad (4.5)$$

$$d_i = \operatorname{argmin}(S_i) \quad (4.6)$$

$$\omega_i = \delta(d_i) \quad (4.7)$$

cuja tendência de sincronização conduz ao consenso das ações em grupos de agentes. O fator  $\frac{1}{2}$  foi incorporado para manter a medida de similaridade  $\cos(\theta_j - \theta_i)$  em um intervalo  $[0, 1]$ . Quando a similaridade das fases  $\cos(\theta_j - \theta_i) \approx 0$ , a variável  $x_j$  não apresenta uma tendência de conflito na solução parcial e, neste caso, considera-se exclusivamente o custo local da função  $f_{ij}$ . Do contrário,  $\cos(\theta_j - \theta_i) \approx 1$  indica uma tendência de conflito na solução parcial e, neste caso, considera-se um fator adicional ao custo local da função  $f_{ij}$ .

**Definição 5.** Processo de sincronização:  $S_i$  representa um vetor com a soma dos custos locais e similaridades de fase sobre uma dada solução parcial contida na visão local  $\text{view}$  do agente, ao considerar  $\forall d \in D_i$  (Equação 4.5). Desta maneira, o novo valor  $d_i$  é escolhido a partir do menor valor contido em  $S_i$ , ao se tratar de um problema de minimização (Equação 4.6). Por fim,  $\omega_i$  representa o custo local  $\delta(d_i)$  quando  $x_i \leftarrow d_i$  (Equação 4.7). Portanto, as ações locais são atribuídas com base na relação entre o custo local e o grau de erro entre as atribuições das variáveis vizinhas contidas na visão local  $\text{view}$ .

## 4.2 COOPT: Coupled Oscillators OPTimization

O algoritmo *Coupled Oscillators OPTimization* (COOPT) introduz conceitos da sincronização física em redes de osciladores acoplados para coordenar as ações dos agentes em um DCOP. Esta estratégia de coordenação objetiva convergir gradualmente o custo local  $\delta(d)$  das variáveis para o melhor custo global  $\delta^*$ , levando em consideração a influência dos agentes vizinhos representada pelas restrições compartilhadas. Desta maneira, cada agente  $a_i$  será influenciado pelos demais vizinhos a escolher um valor  $d$  para sua variável  $x_i$  visando uma aproximação do melhor custo global  $\delta^*$ , ao invés de escolher um valor que melhore exclusivamente o custo local  $\delta(d)$ .

### 4.2.1 Representação do Problema

Como em outros algoritmos para DCOP, o COOPT requer que os agentes sejam priorizados em uma estrutura de pseudoárvore. Nesta estrutura, cada agente possui um pai e múltiplos filhos, representados por  $P_i$  e  $a_j \in C_i$ , respectivamente. Desta forma,  $P_i = \emptyset$  indica que o agente  $a_i$  representa a raiz da pseudoárvore. Por outro lado,  $C_i = \emptyset$  indica que o agente  $a_i$  é uma folha na pseudoárvore. Esta priorização possibilita a definição de um processo monotônico de convergência ao introduzir uma sincronização sequencial entre agentes em um mesmo caminho (ZIVAN; OKAMOTO; PELED, 2014). Além disto, a priorização também permite uma visão local consistente dos agentes em cada ciclo do algoritmo (PERI; MEISELS, 2013).

---

#### Algorithm 1 Geração de pseudoárvore a partir do grafo de restrições

---

```

1: function PSEUDOTREE ▷ dado um grafo não direcionado e conectado  $G(\mathcal{V}, \mathcal{E})$ 
2:    $x_i \leftarrow \text{random}(\mathcal{V})$ 
3:   DEPTHFIRSTSEARCH( $\emptyset, x_i, \emptyset$ )
4: end function
5: function DEPTHFIRSTSEARCH( $p, x_i, V$ )
6:    $P_i \leftarrow p$ 
7:   for each  $\{x_i, x_j\} \in \mathcal{E}$  do
8:      $N_i \leftarrow N_i \cup \{x_j\}$ 
9:     if  $\nexists x_j \in V$  then
10:       $C_i \leftarrow C_i \cup \{x_j\}$ 
11:      DEPTHFIRSTSEARCH( $x_i, x_j, V \cup \{x_i\}$ )
12:     end if
13:   end for
14: end function

```

---

Uma maneira usual para produzir uma pseudoárvore a partir de um DCOP consiste em realizar uma busca em profundidade sobre o grafo de restrições (Algoritmo 1). Nesta técnica, elege-se arbitrariamente um nó do grafo para iniciar a busca (linha 2). Por sua

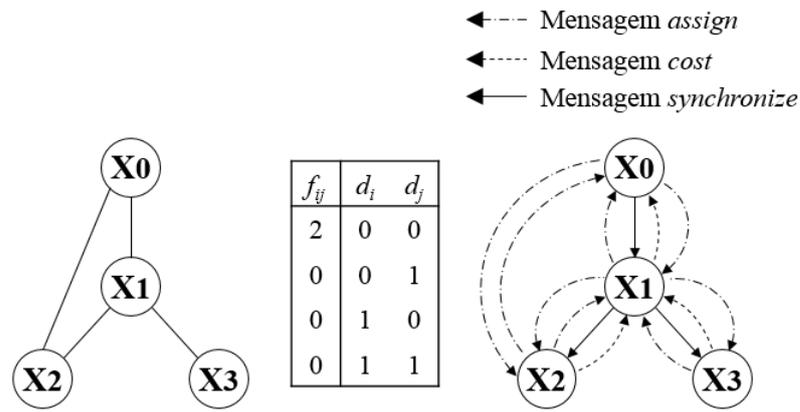


Figura 4.1: Exemplo de DCOP e do modelo de comunicação do COOPT.

vez, o nó eleito, identificado como  $x_i$ , representa a raiz da pseudoárvore, isto é, para este nó  $P_i = \emptyset$  (linhas 3 e 6). Em seguida, são visitados todos os nós vizinhos do nó atual a partir de uma busca em profundidade (linha 7). Os nós vizinhos da variável  $x_i$  atual são identificados pelo conjunto  $N_i$  (linha 8).

A busca em profundidade tem por objetivo armazenar os caminhos percorridos no grafo de restrições a fim de estabelecer relações hierárquicas entre os nós visitados, além de remover os ciclos, quando existentes. Para tal, são percorridos todos os níveis da árvore, armazenando em uma lista temporária  $V$  todos os nós já visitados até o momento, em um dado caminho. Caso algum nó vizinho  $x_j$  ainda não tenha sido visitado durante alguma iteração prévia dentro do mesmo caminho (linha 9), este nó é adicionado ao conjunto  $C_i$ , o qual representa os nós filhos de  $x_i$  (linha 10). Do contrário, o nó vizinho  $x_j$  é considerado um pseudofilho de  $x_i$ . Estes procedimentos são executados recursivamente até que não exista nenhum nó  $x_j$  adjacente ao nó  $x_i$  que ainda não tenha sido visitado (linha 11).

#### 4.2.2 Modelo de Comunicação

O modelo de comunicação do COOPT é composto por três tipos de mensagens, denominados *assign*, *cost* e *synchronize* (Figura 4.1). Estas mensagens determinam o estágio interno dos agentes durante o processo de busca. A mensagem do tipo *assign* objetiva notificar todos os agentes vizinhos, isto é,  $x_j \in N_i$ , sobre um novo valor atribuído à variável  $x_i$  em um determinado momento. Por sua vez, a mensagem do tipo *cost* tem por objetivo informar ao agente vizinho de maior prioridade, isto é, o nó pai  $P_i$ , sobre o custo acumulado na subárvore iniciada em  $x_i$  em um dado momento. Por fim, a mensagem do tipo *synchronize* visa informar quando os agentes de menor prioridade, isto é, os nós filhos  $x_j \in C_i$  de  $x_i$ , devem iniciar um novo estágio de sincronização e, conseqüentemente, escolher um novo valor para suas variáveis.

O COOPT realiza uma busca local *anytime* composta por uma sequência de ciclos de comunicação. Um ciclo de comunicação, denominado *round*, é constituído por múltiplas mensagens do tipo *assign*, *cost* e *synchronize* compartilhadas entre os agentes. Cada agente possui um contador interno, denominado *round*, que incrementa em uma unidade quando um novo estágio de sincronização for iniciado. Assim, o processo de busca de um agente é encerrado quando este contador atinge o  $n$ -ésimo *round* predefinido. Deste modo, é possível determinar previamente o esforço computacional requerido pelo COOPT a partir da quantidade limite de *rounds*.

### 4.2.3 Propriedade *Anytime*

A propriedade *anytime* do COOPT é garantida a partir de uma técnica adaptada do framework *Anytime Local Search DCOP* (ALS\_DCOP) concebido para algoritmos de busca local (ZIVAN; OKAMOTO; PELED, 2014). Este framework descreve um modelo de comunicação que oferece suporte à propriedade *anytime* por meio da priorização dos agentes em uma pseudoárvore. No ALS\_DCOP, as atribuições das variáveis ocorrem sequencialmente em cada caminho da pseudoárvore a partir do nó raiz até os nós folhas. Em seguida são propagados também sequencialmente os custos acumulados em cada subárvore, a partir dos nós folhas até o nó raiz. Ao final do *round* de comunicação, o nó raiz é capaz de computar o custo global acumulado para a solução atual.

**Lema 1.** *Em cada round  $t$  de comunicação do COOPT, os agentes detém uma visão local consistente sobre as atribuições dos agentes vizinhos e o agente raiz é capaz de computar o custo global acumulado da solução completa do  $t$ -ésimo round.*

*Demonstração.* Este Lema será demonstrado por indução a partir da definição recursiva da subárvore iniciada em um dado agente  $a_i$ . Quando o agente  $a_i$  for um nó folha, isto é,  $C_i = \emptyset$ , o custo acumulado na subárvore será igual ao custo local  $\delta(d_i)$ . Por outro lado, quando o agente  $a_i$  não for um nó folha, isto é,  $C_i \neq \emptyset$ , o custo acumulado na subárvore será igual ao custo local  $\delta(d_i)$  acrescido dos custos reportados pelos agentes  $a_j \in C_i$ . Consequentemente, quando a subárvore iniciada no agente  $a_i$  representar a própria estrutura da pseudoárvore, este agente refere-se ao nó raiz e terá condições de calcular o custo global acumulado da solução completa no *round*  $t$ .  $\square$

**Lema 2.** *São mantidas pelos agentes as atribuições que compõem a melhor solução completa encontrada até o  $t$ -ésimo round. Deste modo, quando a busca for encerrada, todos os agentes conhecem a atribuição de sua respectiva variável que resultou no melhor custo global acumulado encontrado.*

*Demonstração.* De acordo com o Lema 1, quando o agente  $a_i$  for o nó raiz da pseudoárvore, isto é, quando  $P_i = \emptyset$ ,  $a_i$  é capaz de calcular o custo global acumulado de uma solução completa de um dado *round*. Portanto, o nó raiz mantém a informação do *round* que resultou no melhor custo global acumulado até o momento. Deste modo, quando o nó raiz finalizar o *round*  $t$ , será comparado o custo global acumulado do *round*  $t$  com o custo global acumulado já conhecido do *round*  $t-1$ . Em seguida, é propagado para todos os nós filhos a informação do *round* que resultou na melhor solução completa até o momento.  $\square$

**Teorema 1.** *A estratégia de busca local COOPT oferece suporte à propriedade anytime a partir de um processo de convergência monotônico que, conseqüentemente, resulta na melhor solução encontrada até o  $t$ -ésimo round.*

*Demonstração.* Este Teorema é consequência direta dos Lemas 1 e 2. Ao final de cada *round*, os agentes conhecem suas atribuições que resultaram na melhor solução completa até o momento. Assim, o COOPT irá seguramente retornar a melhor solução completa encontrada dentro do limite predefinido de  $n$  rounds.  $\square$

#### 4.2.4 Descrição do Algoritmo

Os procedimentos do COOPT encontram-se detalhados no Algoritmo 2. Cada agente  $a_i$  possui um contexto local denominado *context*, o qual representa uma solução parcial no formato  $\{(x_j, d_j, \theta_j), \dots\}$ . O contexto *context* é composto pelo valor corrente  $d$  e fase  $\theta$  das variáveis vizinhas em um determinado *round*. Apenas o agente  $a_i$  controla o estado da variável  $x_i$  e conhece seu domínio  $D_i$ . Além disto, um agente também mantém informações sobre o melhor *round*, melhor atribuição e melhor custo global acumulado em *bestRound*, *bestValue* e *bestCost*, respectivamente.

Ao iniciar o processo de busca do COOPT, é assumido que o grafo de restrições encontra-se representado a partir de uma estrutura de pseudoárvore, conforme descrito pelo Algoritmo 1. A execução do COOPT é iniciada pelo procedimento INITIALIZE, onde cada agente  $a_i$  inicia sua fase  $\theta_i$  com um valor aleatório seguindo uma distribuição gaussiana  $\mathcal{N}(\mu, \sigma^2)$  com média  $\mu = 0$  e desvio padrão  $\sigma^2 = 1$  (linha 2). Cada agente também atribui um valor inicial  $d \in D_i$  para sua variável que resulte exclusivamente no melhor custo local  $\delta(d)$  (linha 3).

Em seguida, ainda no procedimento INITIALIZE, são atribuídas todas as demais informações iniciais de cada agente (linhas 4 a 7) e enviadas mensagens do tipo *assign* para todos os agentes vizinhos  $a_j \in N_i$  (linhas 8 a 9). Estas mensagens visam informar os agentes vizinhos sobre os novos valores  $d_i$  e  $\theta_i$  do agente  $a_i$  no *round* atual. Ao final deste procedimento, quando o agente  $a_i$  for um nó folha, será enviada uma mensagem do tipo

---

**Algorithm 2** Procedimentos do COOPT
 

---

```

1: function INITIALIZE
2:    $\theta_i \sim \mathcal{N}(\mu, \sigma^2)$ 
3:    $d_i, bestValue \leftarrow d \in D_i \mid \min(\delta(d))$ 
4:    $round, bestRound \leftarrow 1$ 
5:    $bestCost \leftarrow \infty$ 
6:    $context, costs \leftarrow \emptyset$ 
7:    $cost, sync \leftarrow \mathbf{false}$ 
8:   for each  $x_j \in N_i$  do
9:     Envia ASSIGN( $x_i, d_i, \theta_i$ ) para  $x_j$ 
10:  end for
11:  if  $C_i = \emptyset$  then
12:     $cost \leftarrow \mathbf{true}$ 
13:    COMPUTECOST
14:  end if
15: end function
16: function ASSIGN( $x_j, d_j, \theta_j$ )  $\triangleright$  ao receber assign
17:    $context \leftarrow context \cup \{(x_j, d_j, \theta_j)\}$ 
18:   if  $cost$  then
19:     COMPUTECOST
20:   end if
21:   if  $sync$  then
22:     PERFORMSYNCH
23:   end if
24: end function
25: function COST( $x_j, \delta_j$ )  $\triangleright$  ao receber cost
26:    $costs \leftarrow costs \cup \{(x_j, \delta_j)\}$ 
27:   if  $\forall x_j \in C_i \mid \exists x_j \in costs$  then
28:      $cost \leftarrow \mathbf{true}$ 
29:     COMPUTECOST
30:   end if
31: end function
32: function SYNCH( $round$ )  $\triangleright$  ao receber synchronize
33:    $sync \leftarrow \mathbf{true}$ 
34:   if  $round \neq bestRound$  then
35:      $bestRound \leftarrow round$ 
36:      $bestValue \leftarrow d_i$ 
37:   end if
38:   PERFORMSYNCH
39: end function
40: function PERFORMSYNCH
41:   if  $\forall x_j \in N_i \mid \exists x_j \in context$  then
42:      $sync \leftarrow \mathbf{false}$ 
43:    $round \leftarrow round + 1$ 
44:   if  $round < n$  then
45:      $S_i \leftarrow \emptyset$ 
46:     for each  $d \in D_i$  do
47:        $\omega \leftarrow \delta(d) + \sum_{x_j \in N_i} f_{ij}(d_i, d_j) \cos(\theta_j - \theta_i)$ 
48:        $S_i \leftarrow S_i \cup \{(d, \omega)\}$ 
49:     end for
50:      $d_i \leftarrow \operatorname{argmin}_{\omega \in S_i}(S_i)$ 
51:      $\omega_i \leftarrow \delta(d_i)$ 
52:      $\theta_i \leftarrow \omega_i + \frac{K}{N} \sum_{x_j \in N_i} f_{ij}(d_i, d_j) \sin(\theta_j - \theta_i)$ 
53:     for each  $x_j \in N_i$  do
54:       Envia ASSIGN( $x_i, d_i, \theta_i$ ) para  $x_j$ 
55:     end for
56:   end if
57:   if  $C_i = \emptyset$  then
58:     if  $round < n$  then
59:       COMPUTECOST
60:     end if
61:   else
62:     for each  $x_j \in C_i$  do
63:       Envia SYNCH( $bestRound$ ) para  $x_j$ 
64:     end for
65:   end if
66: end if
67: end function
68: function COMPUTECOST
69:   if  $\forall x_j \in N_i \mid \exists x_j \in context$  then
70:      $cost \leftarrow \mathbf{false}$ 
71:      $\delta_i \leftarrow \delta(d_i) + \sum_{x_j \in costs} \delta_j$ 
72:     if  $P_i = \emptyset$  then
73:       if  $bestCost > \delta_i$  then
74:          $bestValue \leftarrow d_i$ 
75:          $bestCost \leftarrow \delta_i$ 
76:          $bestRound \leftarrow round$ 
77:       end if
78:       PERFORMSYNCH
79:     else
80:       Envia COST( $x_i, \delta_i$ ) para  $P_i$ 
81:     end if
82:   end if
83: end function

```

---

*cost* para o agente pai  $P_i$ , informando-o sobre o custo da solução parcial da subárvore iniciada em  $a_i$  (linhas 11 a 14).

O procedimento ASSIGN tem por objetivo atualizar as visões locais *context* dos agentes a partir das mensagens do tipo *assign* enviadas pelos vizinhos em um dado *round* (linha 17). Quando um agente atualizar sua visão local com as informações de um dado *round*, podem ser executados os procedimentos COMPUTECOST ou PERFORMSYNCH, de acordo com o estágio interno do agente (linhas 18 a 23). Os procedimentos COMPUTECOST e PERFORMSYNCH são responsáveis por propagar o custo acumulado da solução parcial em uma dada subárvore e iniciar um novo processo de sincronização, respectivamente.

O procedimento COST é executado quando um agente recebe uma mensagem do tipo *cost*. Na sequência, o agente  $a_i$  armazena o custo reportado na mensagem do tipo *cost*, o qual representa o custo acumulado de uma solução parcial contida na subárvore iniciada no remetente da mensagem (linha 26). Quando o agente  $a_i$  receber todos os custos acumulados dos nós filhos no *round* atual, será calculado o custo acumulado da subárvore iniciada em  $a_i$ , como descrito em COMPUTECOST (linha 71). Caso o agente  $a_i$  for a raiz da pseudoárvore, o custo acumulado refere-se ao custo global para a solução completa. Neste caso, são atualizadas as informações do agente raiz  $a_i$  sobre o melhor custo global acumulado, melhor *round* e melhor atribuição (linhas 73 a 77) e iniciado um novo processo de sincronização pelo procedimento PERFORMSYNCH (linha 78).

O procedimento SYNCH é iniciado quando um agente recebe uma mensagem do tipo *synchronize*. Ao receber esta mensagem, o agente  $a_i$  atualiza as informações sobre o melhor *round* e melhor atribuição até o momento (linhas 34 a 37). Quando o agente  $a_i$  possuir a visão local atualizada com as informações de um dado *round*, será iniciado o processo de sincronização descrito pelo procedimento PERFORMSYNCH. Ao iniciar este procedimento, o contador *round* é incrementado em uma unidade e, caso não for atingido o  $n$ -ésimo *round* predefinido, será escolhido um novo valor  $d_i$  para a variável  $x_i$  e atualizadas a frequência  $\omega_i$  e fase  $\theta_i$  do agente  $a_i$  (linhas 43 e 44).

Ainda em PERFORMSYNCH, o agente  $a_i$  calcula a frequência  $\omega$  para cada valor  $d \in D_i$  (Equação 4.5) e adiciona os resultados no vetor  $S_i$  (linhas 46 a 49). Com base nas informações contidas em  $S_i$ , o agente  $a_i$  escolhe um novo valor  $d_i$  que reduz o impacto de sua ação sobre as ações vizinhas e atualiza a frequência  $\omega_i$  com o custo local  $\delta(d_i)$  da nova atribuição (linhas 50 e 51). Na sequência, também é atualizada a fase  $\theta_i$  (Equação 4.1) considerando os novos valores  $\omega_i$  e  $d_i$  (linha 52). Caso o agente  $a_i$  for um nó folha e o  $n$ -ésimo *round* não foi atingido, é executado o procedimento COMPUTECOST (linhas 57 a 60). Caso o agente  $a_i$  não for um nó folha, serão enviadas mensagens do tipo *synchronize* para os agentes filhos com a informação do melhor *round* até o momento (linhas 62 a 64).

#### 4.2.5 Exemplo de Execução

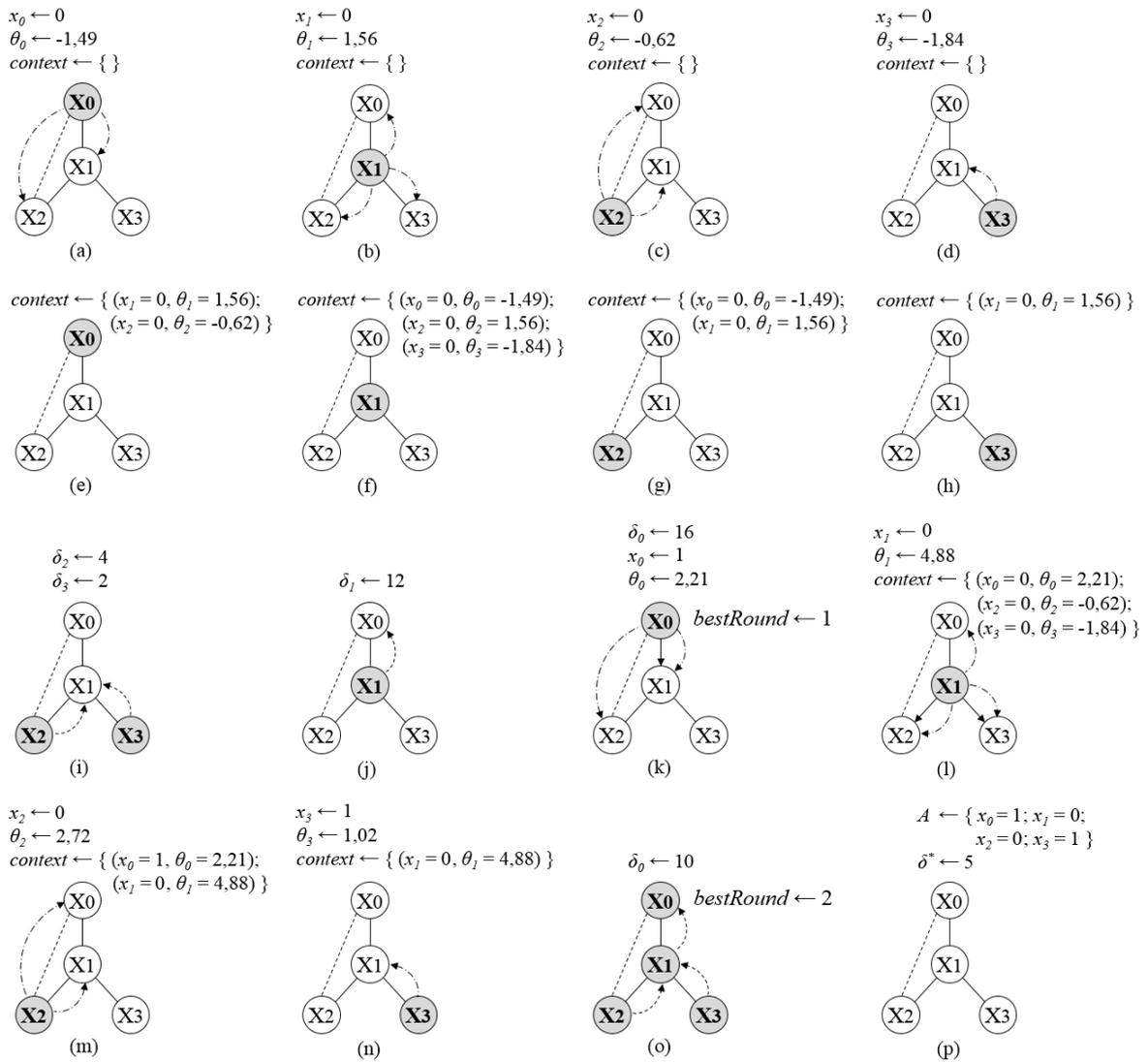


Figura 4.2: Execução do COOPT a partir do problema retratado pela Figura 4.1.

A Figura 4.2 ilustra parte de um exemplo de execução do COOPT (Algoritmo 2) a partir do problema retratado pela Figura 4.1. As linhas pontilhadas da pseudoárvore indicam a relação de pseudopai e pseudofilho, ao passo que as linhas sólidas representam a relação de pai e filho entre os agentes conectados. Devido ao comportamento parcialmente assíncrono do COOPT, será demonstrado uma das possíveis sequências de execução para este problema em dois *rounds* de comunicação. Para este exemplo de execução, foi considerada uma força de acoplamento global  $K = 1$ .

Os agentes iniciam o processo de busca escolhendo concorrentemente valores para suas variáveis que melhorem exclusivamente os custos locais  $\delta(d)$ . Em seguida são iniciados os contextos locais  $context \leftarrow \emptyset$  e fases  $\theta_i \sim \mathcal{N}(0, 1)$  em cada agente. Após isto, todos os agentes enviam mensagens do tipo *assign* para os vizinhos com o propósito de informa-los

sobre os valores iniciais de suas variáveis e fases (Figuras 4.2.a, 4.2.b, 4.2.c e 4.2.d). Estes passos estão descritos no procedimento INITIALIZE.

Na sequência, os agentes processam as mensagens do tipo *assign* recebidas no primeiro *round* e atualizam suas visões locais *context* (Figuras 4.2.e, 4.2.f, 4.2.g e 4.2.h). Este comportamento tem por objetivo assegurar que os agentes possuam uma visão local consistente sobre o *round* atual, a fim de evitar computações locais desnecessárias. Estes passos estão especificados no procedimento ASSIGN (linha 17). Depois dos agentes  $x_2$  e  $x_3$  atualizarem suas visões locais *context*, serão propagados os custos acumulados em suas respectivas subárvores para o agente  $x_1$ .

A propagação dos custos acumulados é realizada por meio de mensagens do tipo *cost*, a partir dos nós folhas da pseudoárvore e em direção ao nó raiz. Portanto, os agentes  $x_2$  e  $x_3$  iniciam este processo reportando para o agente  $x_1$  os custos acumulados  $\delta_2 = 4$  e  $\delta_3 = 2$ , respectivamente (Figura 4.2.i). Por sua vez, o agente  $x_1$  processa as mensagens do tipo *cost* recebidas de  $x_2$  e  $x_3$  e envia para  $x_0$  o custo acumulado  $\delta_1 = 12$  em sua subárvore, considerando o custo local  $\delta(d_i) = 6$  acrescido dos custos acumulados reportados previamente por  $x_2$  e  $x_3$  (Figura 4.2.j). Estes passos estão descritos nos procedimentos COST e COMPUTECOST.

Quando o agente  $x_0$ , isto é, o nó raiz da pseudoárvore, receber a mensagem do tipo *cost* enviada por  $x_1$ , será possível calcular o custo global acumulado  $\delta_0 = 16$  para a solução completa do primeiro *round* (linha 71). O agente  $x_0$  atualiza as informações referentes ao melhor custo  $bestCost \leftarrow 16$ , melhor *round*  $bestRound \leftarrow 1$  e melhor atribuição  $bestValue \leftarrow 0$  (linhas 73 a 77). Neste momento, é encerrado o primeiro *round* de comunicação ao iniciar um novo processo de sincronização no nó raiz  $x_0$  da pseudoárvore (linha 78). Estes passos estão detalhados no procedimento COMPUTECOST.

Em seguida, já no procedimento PERFORMSYNCH, o agente  $x_0$  inicia o segundo *round* de comunicação (linhas 43 e 44), escolhe um novo valor  $d_i \leftarrow 1$  para sua variável e atualiza sua frequência  $\omega_i \leftarrow 2$  e sua fase  $\theta_i \leftarrow 2, 21$ , conforme as Equações 4.5, 4.6 e 4.7 (linhas 46 a 52). O agente  $x_0$  conclui este procedimento ao enviar mensagens do tipo *assign* para  $x_1$  e  $x_2$  sobre os novos valores do segundo *round* (linhas 53 a 55) e ao enviar uma mensagem do tipo *synchronize* para  $x_1$  contendo a informação do melhor *round* até o momento (linhas 62 a 64) (Figura 4.2.k).

O agente  $x_1$  inicia um novo processo de sincronização ao receber as mensagens do tipo *assign* e *synchronize* enviadas por  $x_0$ . Em seguida,  $x_1$  atualiza sua visão local *context* com os novos valores de  $x_0$  (linha 17) e o melhor *round*  $bestRound \leftarrow 1$  (linhas 34 a 37). O agente  $x_1$  inicia o segundo *round* de comunicação (linhas 43 e 44), escolhe um novo valor  $d_i \leftarrow 0$  para sua variável e atualiza sua frequência  $\omega_i \leftarrow 5$  e sua fase  $\theta_i \leftarrow 4, 88$ ,

conforme as Equações 4.5, 4.6 e 4.7 (linhas 46 a 52). O agente  $x_1$  conclui o processo de sincronização ao enviar mensagens do tipo *assign* para  $x_0$ ,  $x_2$  e  $x_3$  sobre os novos valores do segundo *round* (linhas 53 a 55) e ao enviar mensagens do tipo *synchronize* para  $x_2$  e  $x_3$  contendo a informação do melhor *round* até o momento (Figura 4.2.l).

Por sua vez, o agente  $x_2$  inicia um novo processo de sincronização ao receber as mensagens do tipo *assign* enviadas por  $x_0$  e  $x_1$  e ao receber a mensagem do tipo *synchronize* enviada por  $x_1$ . Na sequência, o agente  $x_2$  atualiza sua visão local *context* com os novos valores de  $x_0$  e  $x_1$  (linha 17) e o melhor *round*  $bestRound \leftarrow 1$  (linhas 34 a 37). O agente  $x_2$  inicia o segundo *round* de comunicação (linhas 43 e 44), escolhe um novo valor  $d_i \leftarrow 0$  para sua variável e atualiza sua frequência  $\omega_i \leftarrow 3$  e sua fase  $\theta_i \leftarrow 2,72$ , conforme as Equações 4.5, 4.6 e 4.7 (linhas 46 a 52). O agente  $x_2$  conclui o processo de sincronização ao enviar mensagens do tipo *assign* para  $x_0$  e  $x_1$  (linhas 53 a 55) referentes aos novos valores do segundo *round* (Figura 4.2.m).

De forma similar ao agente  $x_2$ , o agente  $x_3$  inicia o processo de sincronização ao receber as mensagens do tipo *assign* e *synchronize* enviadas por  $x_1$ . O agente  $x_3$  atualiza sua visão local *context* com os novos valores de  $x_1$  (linha 17) e o melhor *round*  $bestRound \leftarrow 1$  (linhas 34 a 37). Em seguida,  $x_3$  inicia o segundo *round* de comunicação (linhas 43 e 44), escolhe um novo valor  $d_i \leftarrow 1$  para sua variável e atualiza sua frequência  $\omega_i \leftarrow 1$  e sua fase  $\theta_i \leftarrow 1,84$ , conforme as Equações 4.5, 4.6 e 4.7 (linhas 46 a 52). O agente  $x_3$  conclui o processo de sincronização ao enviar mensagens do tipo *assign* para  $x_1$  (linhas 53 a 55) sobre os novos valores do segundo *round* (Figura 4.2.n).

Após  $x_2$  e  $x_3$  concluírem o processo de sincronização, os custos acumulados serão propagados novamente até o nó raiz da pseudoárvore (Figura 4.2.o). Deste modo, o agente raiz  $x_0$  será capaz de calcular o custo global acumulado  $\delta_0 = 10$  da solução completa referente ao segundo *round* de comunicação. Depois de completar o segundo *round*, o processo de busca local do COOPT encontrou uma das possíveis soluções ótimas para este problema, onde  $A = \{x_0 \leftarrow 1; x_1 \leftarrow 0; x_2 \leftarrow 0; x_3 \leftarrow 1\}$  com um custo global ótimo  $\delta^* = 5$  (Figura 4.2.p). Estes procedimentos serão repetidos até que o  $n$ -ésimo *round* de comunicação seja atingido.

A Figura 4.3 apresenta a mesma sequência de execução do COOPT ilustrada pela Figura 4.2, no entanto, com ênfase na comunicação realizada entre os agentes. Neste caso, o fluxo de mensagens compartilhadas entre os agentes é representado por meio de um diagrama de sequência, destacando as interações entre os agentes, bem como as informações compartilhadas ao longo do tempo durante os dois primeiros *rounds* da busca. O diagrama de sequência também destaca os momentos em que os agentes executam procedimentos concorrentes durante a busca.

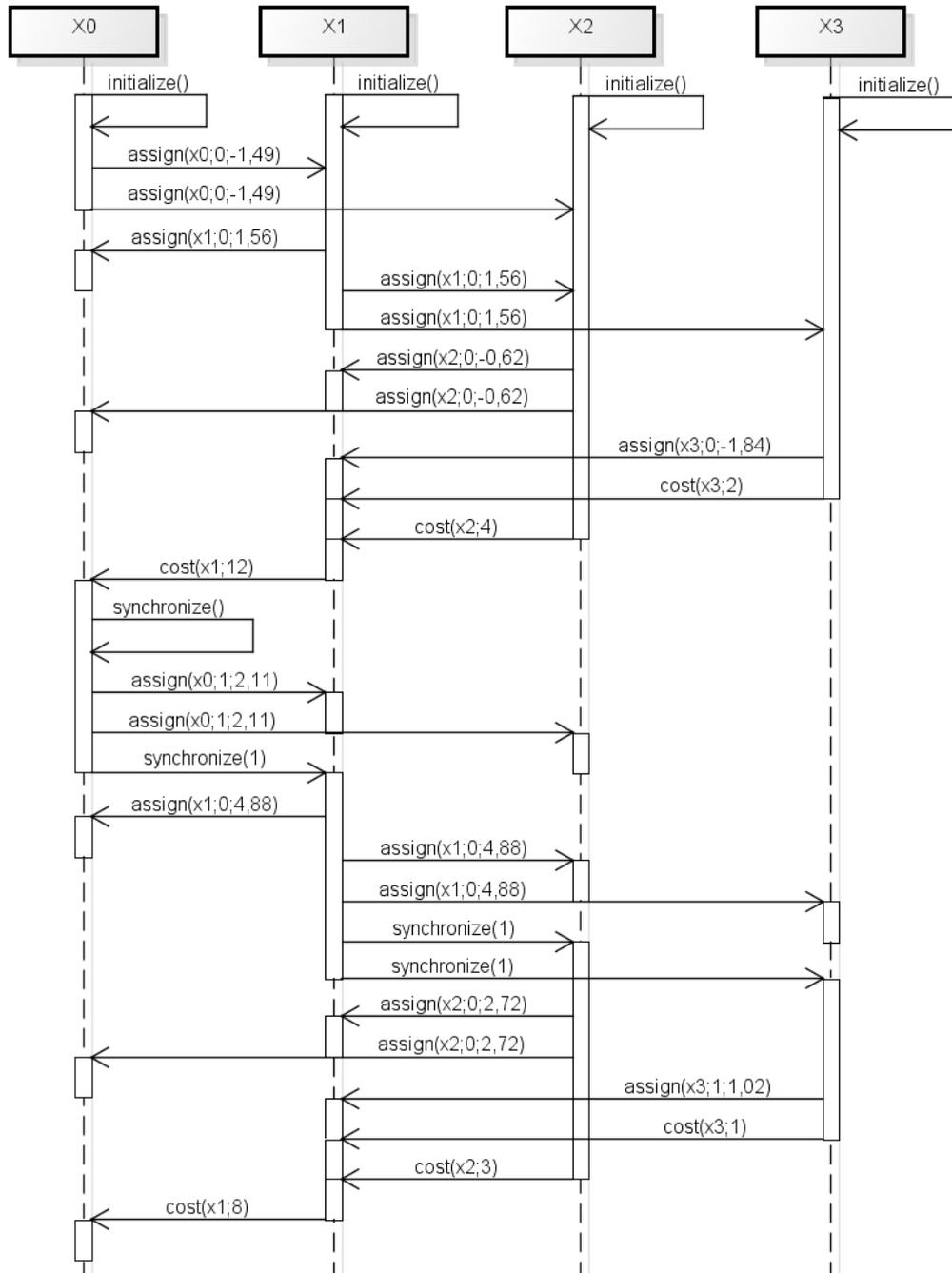


Figura 4.3: Fluxo de mensagens compartilhadas entre os agentes a partir da sequência de execução do COOPT retratada pela Figura 4.2.

Mais especificamente, a partir da Figura 4.3 é possível observar que, para este exemplo de execução, as execuções concorrentes ocorreram entre todos os agentes durante os procedimentos INITIALIZE e ASSIGN. Além disso, ocorreram execuções concorrentes nos agentes  $x_2$  e  $x_3$  durante o procedimento PERFORMSYNCH e COMPUTECOST, considerando que os agentes  $x_2$  e  $x_3$  estão localizados em subárvores distintas na pseudoárvore definida para o problema. Portanto, o COOPT é capaz de realizar computações concorrentes em regiões distintas da pseudoárvore.

#### 4.2.6 Coesão das Fases

O comportamento coletivo emergente do COOPT conduz à formação de grupos de agentes visando o consenso de suas ações individuais. Conforme já mencionado, a discordância entre as ações individuais dos agentes é análoga à fase de um oscilador, cuja tendência de sincronização do COOPT faz com que os agentes sejam atraídos por grupos onde há um consenso sobre a solução parcial. Como consequência, esta tendência de sincronização visa a coesão das fases dos agentes ao longo do tempo.

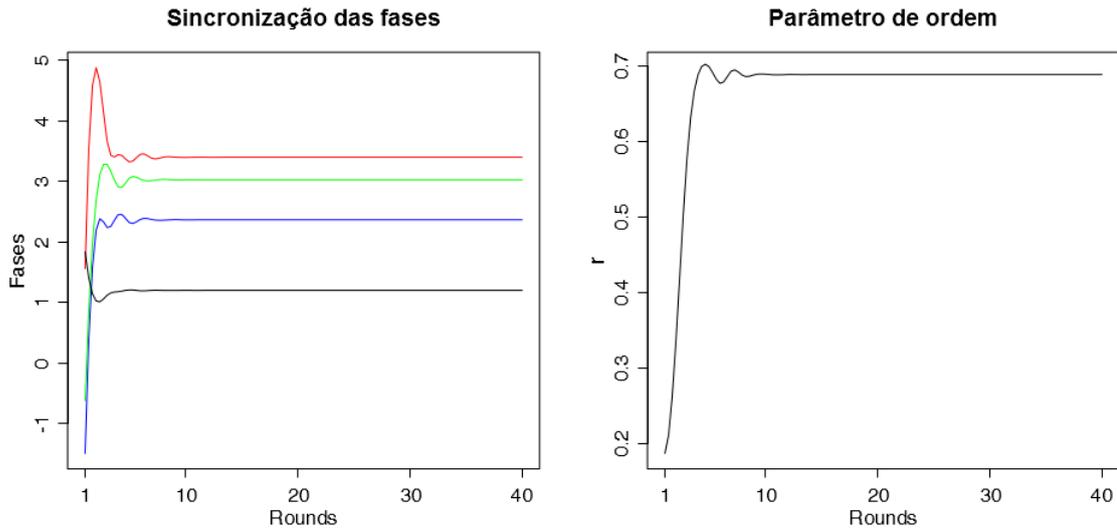


Figura 4.4: Demonstração da tendência de sincronização do COOPT sobre o problema retratado pela Figura 4.2.

A Figura 4.4 descreve a tendência de sincronização e a convergência das fases a partir do problema retratado pela Figura 4.2, considerando um limite de  $n = 40$  rounds. O gráfico à esquerda apresenta a dinâmica da convergência das fases ao longo do tempo. Por sua vez, o gráfico à direita descreve o grau de coesão das fases a partir do parâmetro de ordem  $r$ , de acordo com a Equação 3.5. Ao analisar os gráficos é possível observar uma estabilização na convergência das fases a partir do décimo *round*, aproximadamente.

#### 4.2.7 Análise de Complexidade

Para estimar a complexidade computacional do COOPT, foram avaliados aspectos relacionados ao esforço de comunicação e de processamento local dos agentes. Por se tratar de um algoritmo de busca local, em cada iteração os agentes executam um *round* de comunicação, o qual é constituído por processamentos locais e trocas de mensagens entre os agentes vizinhos. Desta maneira, a análise de complexidade realizada representa o esforço computacional considerando um *round* de comunicação.

#### 4.2.7.1 Comunicação

Como mencionado, o modelo de comunicação do COOPT prevê três tipos distintos de mensagens. O número de mensagens do tipo *synchronize* e *cost* enviadas por *round* é  $2(|\mathcal{V}| - 1)$ , onde  $\mathcal{V}$  representa o conjunto de variáveis do problema. Já as mensagens do tipo *assign* são enviadas  $2|\mathcal{E}|$  vezes por *round*, onde  $\mathcal{E}$  é o conjunto de restrições do problema. Desta maneira, o número de mensagens enviadas por *round* é linear, de acordo com  $\mathcal{O}(2|\mathcal{E}| + 2(|\mathcal{V}| - 1))$ . Esta análise é fundamental para estimar o volume de utilização do canal de comunicação entre os agentes.

Em relação ao tamanho máximo das mensagens, uma mensagem do tipo *assign* contém os valores atuais da variável  $x_i$  e fase  $\theta_i$  do agente  $a_i$ . Por sua vez, uma mensagem do tipo *cost* possui o custo acumulado de uma solução parcial contida na subárvore iniciada no remetente da mensagem. Por fim, uma mensagem do tipo *synchronize* contém a informação do *round* que resultou na melhor solução completa até o momento.

Portanto, o tamanho máximo das mensagens enviadas pelos agentes é constante quando cada agente mantém uma única variável, independentemente da quantidade de variáveis, restrições ou tamanho dos domínios das variáveis em um dado problema. No entanto, em formulações onde os agentes controlam mais de uma variável, o tamanho máximo das mensagens é linear em virtude da quantidade de variáveis mantidas em cada agente. Esta análise contribui para estimar também de uso do canal de comunicação, bem como a capacidade de memória interna requerida pelos agentes para interpretar e processar as mensagens.

#### 4.2.7.2 Processamento Local

Em relação à computação local dos agentes, o número de verificações de restrição é linear por *round*, de acordo com  $\mathcal{O}(2|\mathcal{E}| \max(|D|))$ , onde  $\max(|D|)$  representa o maior domínio  $D \in \mathcal{D}$  entre as variáveis. Em outras palavras, esta análise indica a quantidade de restrições avaliadas pelos agentes em cada *round*, a qual é uma informação essencial para estimar o esforço computacional demandado para resolução de um problema em termos de processamento local dos agentes.

Neste sentido, os fatores que contribuem para o crescimento da complexidade e esforço computacional demandado pelo COOPT são, sobretudo, o número de restrições, o tamanho dos domínios das variáveis e o limite de *rounds* predefinidos. Diante desta análise de complexidade, torna-se evidente a principal vantagem dos algoritmos *anytime* como o COOPT, os quais oferecem flexibilidade sobre o tempo e recursos computacionais

demandados pelo processo de busca por meio de um limite  $n$  de iterações predefinido. Esta característica pode ser determinante para escolha do algoritmo para DCOP em cenários complexos e em larga escala.

## 4.3 Implementação do COOPT

A implementação do algoritmo COOPT foi realizada em um framework para DCOP, denominado FRODO<sup>1</sup> (LÉAUTÉ; OTTENS; SZYMANEK, 2009), distribuído sob licença GNU Affero GPL<sup>2</sup>. O framework FRODO é uma plataforma popular, distribuída e de código aberto para simulação de DCOP baseada na tecnologia Java<sup>3</sup>.

Além do ambiente de simulação, o framework FRODO também dispõe de diversas métricas para avaliação de desempenho dos algoritmos disponíveis em sua infraestrutura básica. Um dos principais benefícios deste framework é a sua arquitetura modular e extensível, possibilitando a definição de novos algoritmos, métricas para avaliação de desempenho, representação de diferentes classes de problemas e até mesmo variações do formalismo DCOP.

### 4.3.1 Plataforma de Simulação FRODO

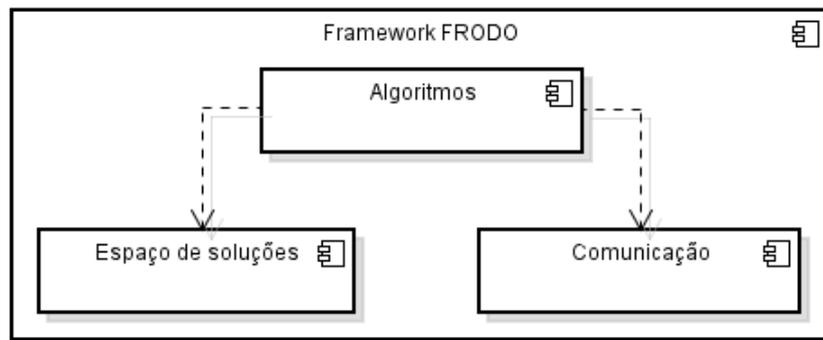


Figura 4.5: Principais camadas da arquitetura modular do FRODO. Adaptada de (LÉAUTÉ; OTTENS; SZYMANEK, 2009).

A Figura 4.5 ilustra as três principais camadas extensíveis e suas dependências, disponíveis na arquitetura modular da plataforma FRODO. A camada *Algoritmos* possui um conjunto de classes que descrevem os comportamentos dos algoritmos suportados pelo framework FRODO. Já a camada *Comunicação* é constituída por um conjunto de classes que representam estruturalmente as mensagens de cada algoritmo, além de possuir

<sup>1</sup>Disponível em <https://sourceforge.net/projects/frodo2/>

<sup>2</sup>Mais informações em <https://www.gnu.org/licenses/agpl-3.0.html>

<sup>3</sup>Disponível em <http://docs.oracle.com/javase/8/>

também a implementação de diversas métricas padrão para avaliação de desempenho sobre a perspectiva de comunicação. Por fim, a camada *Espaço de soluções* contém as classes que representam um subproblema e o hiperplano do espaço de soluções.

Deste modo, para implementação do COOPT no framework FRODO, foi necessário o desenvolvimento de quatro novas classes nas camadas *Algoritmos* e *Comunicação*, a saber, `frodo.agents.COOPT`, `frodo.agents.AssignMessage`, `frodo.agents.CostMessage` e `frodo.agents.SyncMessage`. A classe `COOPT` descreve o comportamento de um agente no COOPT, ao passo que as classes `AssignMessage`, `CostMessage` e `SyncMessage` representam estruturalmente as mensagens requeridas pelo modelo de comunicação do algoritmo.

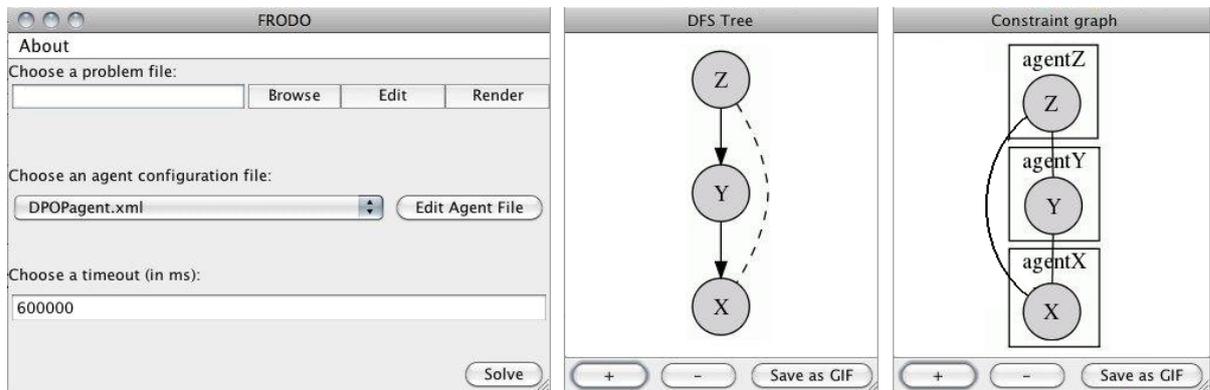


Figura 4.6: Interface gráfica da plataforma FRODO.

A plataforma FRODO também dispõe de uma interface gráfica para configuração e execução do ambiente de simulação (Figura 4.6). Nesta interface é necessário especificar o problema a ser resolvido, a configuração dos agentes e um tempo limite para o processo de busca. Um problema é representado por um documento descrito em linguagem de marcação XML<sup>4</sup>. A configuração dos agentes também é representada por um documento XML, contudo, esta configuração possui a descrição do comportamento dos agentes, bem como as mensagens requeridas pelo modelo de comunicação do algoritmo. Por padrão, a plataforma FRODO oferece suporte aos principais algoritmos para DCOP completos e incompletos existentes na literatura.

### 4.3.2 Representação do Problema

Como mencionado anteriormente, um DCOP é representado na plataforma FRODO por meio de um documento XML. Este documento descreve os componentes envolvidos em um DCOP, tais como agentes, variáveis, domínios e restrições. A Figura 4.7 ilustra

<sup>4</sup>Mais detalhes em <https://www.w3.org/XML/>

```

1 <instance>
2   <presentation name="coloracao-grafo" maxConstraintArity="2"
3     maximize="false" format="XCSP_2.1_FRODO"/>
4   <agents nbAgents="3">
5     <agent name="agentX"/>
6     <agent name="agentY"/>
7     <agent name="agentZ"/>
8   </agents>
9   <domains nbDomains="1">
10    <domain name="cores" nbValues="3">1..3</domain>
11  </domains>
12  <variables nbVariables="3">
13    <variable name="X" domain="cores" agent="agentX"/>
14    <variable name="Y" domain="cores" agent="agentY"/>
15    <variable name="Z" domain="cores" agent="agentZ"/>
16  </variables>
17  <relations nbRelations="1">
18    <relation name="dif" arity="2" nbTuples="3"
19      semantics="soft" defaultCost="0">
20      infinity: 1 1|2 2|3 3
21    </relation>
22  </relations>
23  <constraints nbConstraints="3">
24    <constraint name="X-Y" arity="2" scope="X_Y" reference="dif"/>
25    <constraint name="X-Z" arity="2" scope="X_Z" reference="dif"/>
26    <constraint name="Y-Z" arity="2" scope="Y_Z" reference="dif"/>
27  </constraints>
28 </instance>

```

Figura 4.7: Exemplo de um DCOP representado em formato XML no FRODO

a representação de um problema de coloração de grafos como um DCOP. Este exemplo descreve um problema envolvendo três agentes, três variáveis e três restrições, onde as funções de custo visam evitar que variáveis vizinhas possuam estados idênticos.

No exemplo da Figura 4.7, a marcação `<agents>` define os agentes envolvidos em um DCOP. Já a marcação `<domains>` descreve os domínios das variáveis, isto é, os possíveis estados que cada variável pode assumir. O FRODO suporta tanto domínios heterogêneos quanto homogêneos em um DCOP. Por sua vez, a marcação `<variables>` descreve as variáveis em um DCOP, especificando o domínio e o agente que irá controlar cada variável. A marcação `<relations>` define o comportamento das relações lógicas, isto é, as funções de custo, entre as variáveis. No exemplo há apenas uma função de custo utilizada em três restrições distintas.

Com relação às funções de custo, é possível caracterizá-las como *soft* ou *hard* pelos atributos `semantics="soft"` e `semantics="hard"` respectivamente. Esta definição dependerá da natureza das funções de custo para cada instância de problema. Também é possível definir um valor padrão para uma função de custo pelo atributo `defaultCost`, o qual será assumido quando não houver uma combinação previamente especificada no conteúdo da marcação `<relation>`.

Por fim, a marcação `<constraints>` é utilizada para descrever todas as restrições em um DCOP. Para tal são utilizadas as marcações `<constraint>` para especificar as variáveis envolvidas, bem como a função de custo que será aplicada sobre cada restrição do problema. As variáveis envolvidas em uma restrição são especificadas pelo atributo `scope`, ao passo que a função de custo a ser aplicada sobre uma restrição é referenciada pelo atributo `reference`.

### 4.3.3 Representação de um Agente

A configuração dos agentes na plataforma FRODO também é realizada por meio de um documento XML. Neste caso, o documento descreve o algoritmo a ser utilizado e as mensagens suportadas pelo agente, bem como as configurações para ativar a medição do desempenho durante as simulações. Neste documento também é possível configurar os parâmetros adicionais de um algoritmo para DCOP em particular.

A Figura 4.8 ilustra o exemplo de configuração de um agente para o algoritmo COOPT. A marcação `<agentDescription>` contém informações básicas de um agente, tais como a classe utilizada para gerenciar a fila de mensagens dos agentes (atributo `className`) e os parâmetros para ativar a medição do tempo de execução e a coleta de informações relativas à comunicação entre agentes (atributos `measureTime` e `measureMsgs`).

```

1 <agentDescription className="frodo2.algorithms.SingleQueueAgent"
2     measureTime="true"
3     measureMsgs="true" >
4   <mailman mailmanClass="frodo2.communication.mail.CentralMailer"/>
5   <parser parserClass="frodo2.algorithms.XCSPparser"
6     publicAgents="true"
7     domClass="frodo2.solutionSpaces.AddableInteger"
8     utilClass="frodo2.solutionSpaces.AddableInteger"
9     countNCCCs="true"/>
10  <modules>
11    <module className="frodo2.agents.coopt.COOPT"
12      reportStats="true">
13      <messages>
14        <message name="ASSIGN_MSG_TYPE"
15          value="ASSIGN_MSG"
16          ownerClass="frodo2.agents.coopt.AssignMessage"/>
17        <message name="COST_MSG_TYPE"
18          value="COST_MSG"
19          ownerClass="frodo2.agents.coopt.CostMessage"/>
20        <message name="SYNCHRONIZE_MSG_TYPE"
21          value="SYNCHRONIZE_MSG"
22          ownerClass="frodo2.agents.coopt.SyncMessage"/>
23      </messages>
24    </module>
25  </modules>
26 </agentDescription>

```

Figura 4.8: Estrutura de um agente representado em formato XML no FRODO

O tempo de execução é medido em milissegundos considerando todo o ciclo de vida dos agentes. Já as informações coletadas sobre a comunicação entre os agentes possibilita medir a quantidade total de mensagens enviadas, o total de informações compartilhadas entre os agentes e o tamanho da maior mensagem enviada.

Por sua vez, a marcação `<mailman>` descreve pelo atributo `mailmanClass` a classe utilizada para representar a fila de mensagens de um agente. Já a marcação `parser`, por meio do atributo `parserClass`, especifica a classe que será utilizada para interpretar o documento XML de um dado DCOP. Esta marcação também especifica as classes que representam o domínio das variáveis e a utilidade das soluções pelos atributos `domClass` e `utilClass` respectivamente. Todas as classes mencionadas neste exemplo até então são disponibilizadas por padrão no framework FRODO.

Além disto, na marcação `parser` também é possível ativar a contagem do número de vezes em que as funções de custo são avaliadas, por meio do atributo `countNCCCs`. Por fim, a marcação `<modules>` descreve o comportamento do agente para um determinado algoritmo. No exemplo, é especificado pelo atributo `className` qual é a classe que implementa o comportamento do algoritmo COOPT. O atributo `reportStats` é utilizado para exibir as estatísticas de desempenho ao final de cada simulação. Já as classes que

representam as mensagens do modelo de comunicação do COOPT estão especificadas na marcação `messages` pelo atributo `ownerClass`.

#### 4.3.4 Arquitetura da Implementação

Como já mencionado, para a implementação do algoritmo COOPT na plataforma FRODO foi necessária a adição de três novas classes na infraestrutura padrão. A Figura 4.9 ilustra os detalhes desta implementação no FRODO a partir de um diagrama de classe. Além das classes adicionadas para o COOPT, o diagrama também destaca as principais classes da plataforma FRODO. A classe `COOPT` estende a classe genérica `Agent`, utilizada para definição de um agente na plataforma FRODO. Em outras palavras, a classe `Agent` deve ser complementada estruturalmente com informações e comportamentos específicos de um agente do COOPT.

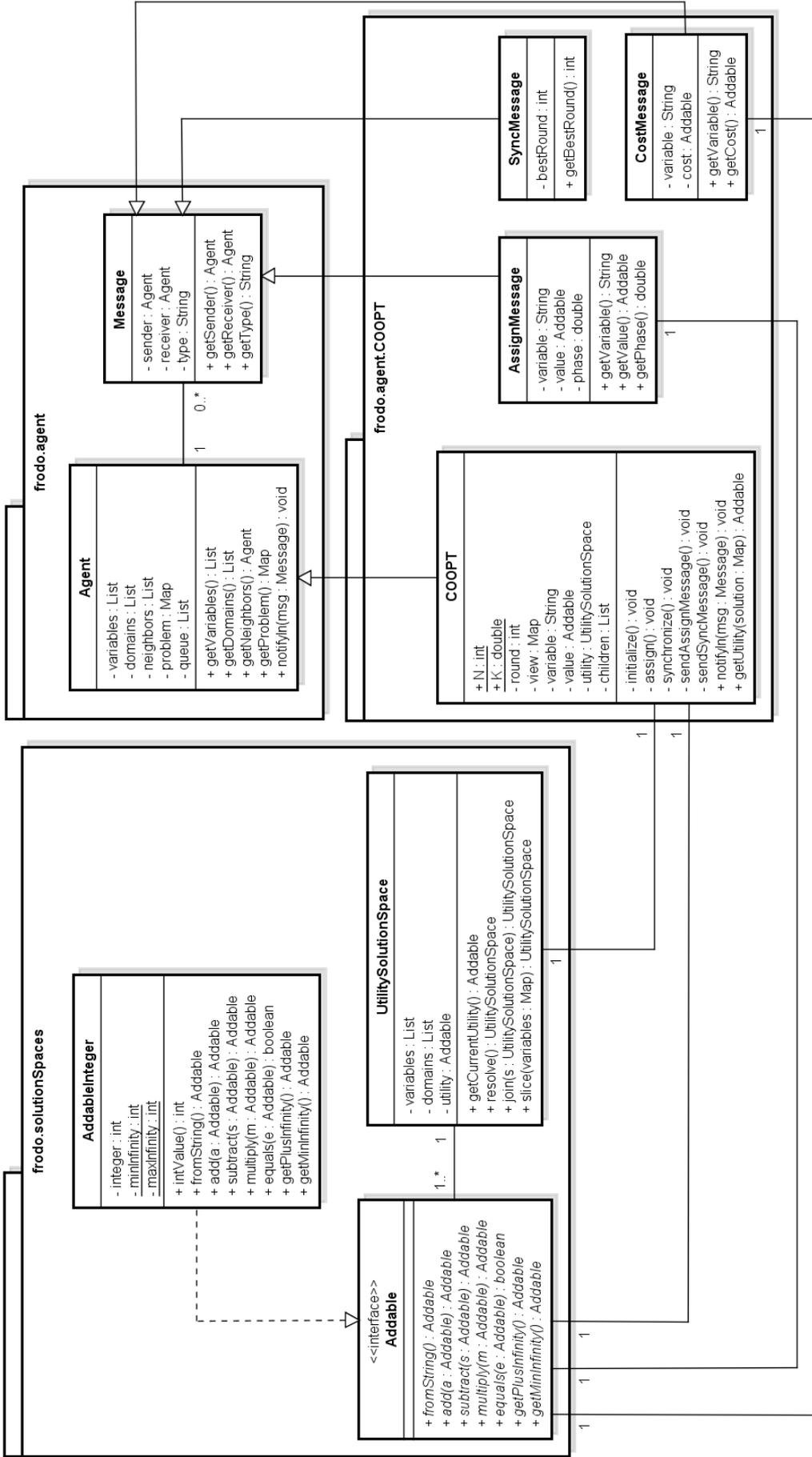


Figura 4.9: Diagrama de classe da implementação do COOPT no framework FRODO.

Neste sentido, a classe genérica **Agent** fornece somente uma infraestrutura básica para definição de um agente no FRODO, tais como a representação do subproblema, variáveis, restrições, agentes vizinhos e o gerenciamento da fila de mensagens. Já as classes **AssignMessage**, **CostMessage** e **SyncMessage** representam as mensagens do tipo *assign*, *cost* e *synchronize* do COOPT, respectivamente. Estas as classes **AssignMessage**, **CostMessage** e **SyncMessage** herdam a estrutura da classe **Message** do framework FRODO, cuja responsabilidade consiste em, sobretudo, fornecer estatísticas provenientes da camada de comunicação dos algoritmos.

Por sua vez, a interface **Addable** também pertence à infraestrutura básica do FRODO e descreve de maneira abstrata os possíveis estados de uma variável. Esta classe também é utilizada para descrever a representação do produto de uma função de custo. Já a classe **AddableInteger** representam os estados das variáveis e o resultado das funções de custo por meio de números inteiros não negativos. Finalmente, a classe **UtilitySolutionSpace** descreve um possível subespaço do problema. A partir desta representação de um subespaço, é possível calcular a utilidade de uma solução parcial para um determinado problema.

## 4.4 Considerações Finais

Neste capítulo foi apresentado um método para resolução de DCOP baseado em conceitos da sincronização física em redes de osciladores acoplados. O propósito para utilização desta fundamentação teórica tem por objetivo capturar as propriedades de convergência e escalabilidade tipicamente observadas em modelos de sincronização física e introduzi-las em novas estratégias para resolução de DCOP. Mais especificamente, o método para resolução de DCOP apresentado, denominado COOPT, utiliza o modelo de Kuramoto para reduzir iterativamente o grau de divergência local das soluções parciais por meio de acoplamentos senoidais entre os elementos da rede.

Para implementação do COOPT foi utilizado um framework popular para DCOP, denominado FRODO. A versão do COOPT apresentado por este capítulo requer alguns pré-requisitos, tais como: i) cada agente deve controlar uma única variável; ii) é necessário definir previamente um parâmetro inicial para o algoritmo, o qual representa uma força de acoplamento global entre os agentes; e iii) o problema deve ser representado em uma estrutura de pseudoárvore. Espera-se que estas limitações, sobretudo os itens i e ii, sejam endereçadas em pesquisas futuras a fim de aumentar a aplicabilidade do método proposto.

# Capítulo 5

## Avaliação Experimental

Uma das principais motivações para concepção do método para resolução de DCOP proposto por este trabalho reside, sobretudo, na limitação dos algoritmos atuais ao se tratar de problemas envolvendo ambientes complexos e em larga escala. Estes tipos de ambientes comumente requerem algoritmos distribuídos e escaláveis que sejam capazes de lidar com severas restrições de comunicação e computação.

Neste sentido, é imprescindível que a avaliação experimental do método proposto aborde questões inerentes a este domínio de aplicação. O presente capítulo visa apresentar a metodologia de avaliação da proposta para este trabalho e os resultados obtidos pelos experimentos realizados, bem como uma discussão final sobre as oportunidades e pontos de atenção observados sobre o COOPT.

### 5.1 Metodologia de Avaliação

A avaliação experimental deste trabalho foi realizada a partir de resultados obtidos por instâncias de DCOP geradas aleatoriamente. No entanto, o processo de geração dos problemas introduziu diferentes níveis de complexidade ao considerar variações no número de agentes envolvidos, densidade do grafo de restrições e diferentes topologias de rede. Diversos estudos (GASTON; DESJARDINS, 2005; LE et al., 2015, 2016) têm demonstrado que a topologia da rede pode apresentar um impacto significativo durante a resolução distribuída do problema. Portanto, a avaliação experimental comparou os resultados obtidos sobre diferentes topologias de rede, com o propósito de identificar as condições mais favoráveis entre os algoritmos utilizados na avaliação.

### 5.1.1 Protocolo de Avaliação

Para avaliar o desempenho do COOPT, foram realizados diversos testes estatísticos a fim de comparar os resultados obtidos com outros algoritmos completos e incompletos que compõem o estado da arte sobre DCOP. Os algoritmos utilizados na comparação foram o DPOP, DSA, MGM, MGM-2 e o DUCT, mencionados anteriormente na Seção 2.7.1. Devido à natureza não determinística de alguns algoritmos avaliados e ao processo de geração aleatória das instâncias de DCOP utilizado pelos experimentos, infelizmente não é possível determinar com precisão se os resultados encontram-se em uma distribuição normal. Em casos como este recomenda-se o uso de testes estatísticos não paramétricos, pois são menos restritivos que os testes paramétricos e apresentam uma menor dependência de hipóteses.

Portanto, a comparação dos resultados obtidos em cada algoritmo foi realizada por meio de testes estatísticos não paramétricos de Friedman (FRIEDMAN, 1940) e de Nemenyi (NEMENYI, 1963). O teste de Friedman possibilita identificar se há significância estatística entre os resultados obtidos, mesmo em condições não paramétricas. Sob hipótese nula  $H_0$ , o teste de Friedman afirma que todos os algoritmos possuem desempenho equivalente, ao passo que a rejeição indica a existência de diferença estatisticamente significativas entre eles.

O teste de Friedman é realizado a partir de uma classificação ordinal (*ranking*) dos resultados de cada algoritmo. Mais especificamente, a estatística do teste é calculada por  $Q = \frac{12}{mk(k+1)} \sum_{j=1}^k R_j^2 - 3m(k+1)$ , onde  $m$  é o número total de instâncias,  $k$  é o número de graus de liberdade e  $R_j^2$  é o quadrado do total de *rankings* para cada algoritmo. O teste rejeita  $H_0$  se  $Q$  for maior do que um determinado nível de confiança  $\alpha$ . Depois de identificar uma diferença estatística entre os resultados obtidos, é possível utilizar um teste *post-hoc*, como o teste de Nemenyi, por exemplo, para fornecer uma classificação de desempenho dos algoritmos avaliados.

Por sua vez, o teste *post-hoc* de Nemenyi produz uma classificação ascendente a partir de testes em pares entre todas as variáveis. Esta classificação é gerada por uma distância crítica ( $|\bar{R}_i - \bar{R}_j|$ ) para cada par  $i$  e  $j$  de algoritmos avaliados, a partir um determinado nível de significância. A distância crítica para o teste de Nemenyi é calculada por  $CD = \frac{q_\alpha}{\sqrt{2}} \sqrt{\frac{k(k+1)}{6m}}$ , onde  $m$  é o número total de instâncias,  $k$  é o número de graus de liberdade e  $q_\alpha$  é o nível de confiança.

### 5.1.2 Categoria de Problemas

Como mencionado, a avaliação experimental do COOPT foi realizada a partir de instâncias aleatórias de DCOP. A plataforma FRODO dispõe de um gerador de problemas aleatórios, dentre outras categorias de DCOP. Os parâmetros de entrada requeridos para este gerador de DCOP são: i) número de variáveis; ii) densidade do grafo; e iii) *tightness*. A densidade de um grafo é uma razão entre o número de arestas do grafo em relação ao grafo completo com o mesmo número de nós (WASSERMAN; FAUST, 1994). Esta informação é expressa no FRODO por um número real entre 0 e 1, onde 0 representa um grafo totalmente desconexo e 1 representa um grafo completo. Em seguida, o número de arestas resultantes é distribuído aleatoriamente entre pares de variáveis em um determinado problema, as quais representam o tamanho do problema.

Por sua vez, o *tightness* refere-se a um nível de restrição das funções de custo, isto é, esta propriedade indica o quão restritivas são as combinações de valores que resultam em violação de restrição (GRINSHPOUN et al., 2013). O nível de restrição também é expresso no FRODO por um número real entre 0 e 1, onde 0 produz funções de custo sem relevância alguma para o problema e 1 produz funções que, dada qualquer combinação de valores, a restrição sempre será violada. A variação de *tightness* pode gerar impacto sobre o esforço computacional requerido pelos algoritmos, além da qualidade das soluções encontradas em algoritmos incompletos.

### 5.1.3 Topologias Avaliadas

O processo de geração aleatória de instâncias de DCOP produziu problemas mais realistas ao considerar as topologias predominantemente utilizadas para descrever modelos de redes complexas e em larga escala em sistemas multiagente, denominadas: i) redes regulares; ii) redes aleatórias; iii) redes de mundos pequenos; e iv) redes livre de escala (SCERRI; VINCENT; MAILLER, 2010). Estas topologias são produzidas a partir de diferentes padrões de conexão entre os elementos e dinâmicas de evolução de uma rede, conforme já abordado previamente pela Seção 3.3 deste trabalho.

No entanto, ao se tratar de redes complexas, existem diversas variações dos modelos de redes aleatórias, mundos pequenos e livres de escala disponíveis na literatura. Desta maneira, o modelo de redes aleatórias utilizado na avaliação experimental é conhecido como Erdős-Rényi (ER) (ERDÖS; RÉNYI, 1959). Já o modelo de redes de mundos pequenos utilizado é denominado Watts-Strogatz (WS) (WATTS; STROGATZ, 1998). Por sua vez, para as redes livres de escala foi adotado o modelo chamado Barabási-Albert (BA) (BARABÁSI; ALBERT, 1999).

Em especial, os modelos utilizados de redes de mundos pequenos e redes livres de escala requerem dois parâmetros iniciais em particular, além do número de nós e de arestas do grafo. Estes parâmetros particulares referem-se à probabilidade  $p$  de reconexões em redes de mundos pequenos e a função de potência  $\gamma$  de conexões preferenciais em redes livres de escala. Desta maneira, para a avaliação experimental deste trabalho, foram considerados os parâmetros  $p = 0,2$  de probabilidade de reconexões e potência  $\gamma = 1$  para conexões preferenciais lineares, respectivamente.

Deste modo, a avaliação experimental levou em consideração diferentes níveis de complexidade nos problemas gerados aleatoriamente a partir de quatro propriedades distintas: i) número de variáveis; ii) densidade da rede; iii) *tightness*; e iv) topologia da rede. A geração aleatória das redes, conforme as topologias utilizadas nos experimentos, foi realizada por meio do pacote de extensão *igraph*<sup>1</sup> do software de computação estatística R<sup>2</sup> (R Development Core Team, 2008). Mais especificamente, as redes complexas foram produzidas no software R e posteriormente convertidas para o padrão de representação de DCOP no FRODO, conforme descrito na Seção 4.2.1.

#### 5.1.4 Geração dos Problemas

A fim de assegurar a confiabilidade dos experimentos e resultados obtidos, foram geradas 30 instâncias de problemas a partir da permutação simples das propriedades de rede citadas previamente, isto é, número de variáveis, densidade, *tightness* e topologia de rede. Além disto, para cada instância de problema, foram considerados os resultados médios de 30 execuções para cada algoritmo avaliado. A utilização do resultado médio justifica-se pelo fato da maioria dos algoritmos avaliados não serem determinísticos, produzindo mais de uma possível sequência de execução e, por consequência, diferentes resultados, sobre o mesmo problema.

Mais especificamente, os experimentos envolveram problemas cujo tamanho variou entre 10 e 200 variáveis, densidade entre 30% e 90%, *tightness* entre 30% a 90% e um domínio homogêneo para as variáveis, onde  $|D| = 4$ . Além disto, o COOPT, assim como o DSA, MGM e MGM-2 implementam uma busca local e requerem a definição de um limite  $n$  de *rounds* (ZHANG et al., 2005; PEARCE; TAMBE; MAHESWARAN, 2008). Portanto, os experimentos consideraram um limite de  $n = 50$  *rounds* para os algoritmos de busca local avaliados. A Tabela 5.1 apresenta as propriedades de rede utilizadas na geração aleatória dos problemas.

<sup>1</sup>Disponível em <http://igraph.org/r/>

<sup>2</sup>Disponível em <https://www.r-project.org/>

Tabela 5.1: Parâmetros utilizados para geração aleatória dos problemas.

Topologias de rede	Variáveis	Domínio das variáveis	Restrições	% <i>Tightness</i>
Regular	10	[0..3]	30%	30%
Aleatória	50		50%	50%
Mundos pequenos	100		90%	90%
Livre de escala	200			

## 5.2 Métricas Utilizadas

Sobre as métricas utilizadas para a avaliação experimental do COOPT, foram consideradas: i) quantidade de ciclos de mensagens; ii) quantidade de verificações não concorrente de restrições (NCCC); iii) tamanho da maior mensagem enviada; iv) volume de informações compartilhadas entre os agentes (*throughput* de rede); e iv) qualidade das soluções encontradas. Estas métricas permitem mensurar o esforço computacional, o uso do canal de comunicação e a quantidade de memória interna dos agentes requerida pelos algoritmos. Estas métricas são disponibilizadas por padrão no ambiente de simulação do FRODO, quando os agentes estão configurados com os parâmetros `countNCCCs ← true` e `measureMsgs ← true`, respectivamente.

### 5.2.1 Qualidade das Soluções

Devido ao fato do COOPT, DSA, MGM, MGM-2 e DUCT serem métodos de busca incompletos, a qualidade das soluções encontradas é um critério indispensável para compor a avaliação de desempenho dos algoritmos (PETCU, 2007). A qualidade das soluções é tipicamente descrita como utilidade, quando o problema de otimização se refere à maximização das funções de recompensas. Já em problemas de minimização, a qualidade das soluções é retratada como custo das restrições. Os problemas utilizados nesta avaliação experimental referem-se à problemas de minimização, isto é, a qualidade das soluções é denominada custo.

Outro critério comumente utilizado para avaliação de desempenho em algoritmos incompletos para DCOP consiste em medir a distância entre a qualidade de uma solução ótima para o problema e a qualidade das soluções sub-ótimas encontradas (PETCU, 2007). No entanto, não foi possível medir a distância entre as soluções ótimas em todos os cenários avaliados devido ao nível de complexidade dos problemas utilizados nos experimentos e o esforço computacional requerido pelos algoritmos completos.

### 5.2.2 Ciclos de Mensagens

Ciclos de mensagens possibilitam avaliar a eficiência do modelo de comunicação de um algoritmo com base na quantidade de mensagens compartilhadas entre os agentes (MODI et al., 2005). Um ciclo de mensagem representa o tempo demandado por um agente para processar uma mensagem contida em sua fila de mensagens e respondê-la aos agentes relacionados (YEOH, 2010). Para calcular os ciclos de mensagens requeridos em um dado problema, cada agente possui um contador *cycles* que inicia com 0 e é incrementado com  $t$  toda vez que um agente recebe uma mensagem.

Deste modo, um incremento  $t > 0$  representa o tempo médio de cada agente para processar as mensagens recebidas, isto é, levando em consideração um tempo previamente definido para um agente interpretar cada mensagem e realizar alguma ação adiante. Não obstante, quando utilizado um incremento  $t = 1$ , os ciclos de mensagens representam simplesmente a quantidade de mensagens recebidas pelos agentes, sendo este o valor utilizado na avaliação experimental deste trabalho. Esta métrica é adequada para avaliar o desempenho de algoritmos em cenários cuja comunicação é mais restritiva do que a capacidade de processamento local de cada agente.

### 5.2.3 NCCC

Por outro lado, a quantidade de verificações não concorrente de restrições, ou *Non-Concurrent Constraint Check* (NCCC), objetiva medir o volume de processamento local do agente em relação ao número de vezes que as funções de custo são avaliadas (MEISELS et al., 2002). Para tal, cada agente mantém um contador  $NCCC^a$  que inicia com 0. Durante o processo de execução do algoritmo, este contador irá incrementar em todo momento que as funções de custo forem avaliadas durante o processo de busca, isto é,  $NCCC^a \leftarrow NCCC^a + 1$ .

O NCCC também pode incrementar quando uma mensagem for enviada por um agente, levando em consideração um critério  $t$  que representa a confiabilidade da rede. Assim, quando um agente  $a_i$  receber uma mensagem de  $a_j$ , seu contador irá incrementar conforme  $NCCC^{a_i} \leftarrow \max(NCCC^{a_i}, NCCC^{a_j} + t)$ . Neste caso, o fator  $t = 0$  representa uma rede confiável, ao passo que  $t > 0$  indica um certo grau de latência da rede. Para os experimentos deste trabalho foi utilizado o fator  $t = 0$ . Esta métrica é adequada para avaliar o processamento local concorrente entre os agentes durante a busca.

### 5.2.4 Tamanho das Mensagens e Informações Compartilhadas

O tamanho das mensagens possibilita estimar o volume de informação que o agente precisa manter em sua memória interna para enviar, receber ou processar uma mensagem (MEISELS et al., 2002). Esta métrica é relevante pois é comum os agentes apresentarem severas limitações dos recursos computacionais, incluindo a capacidade de armazenamento temporário de dados. Alguns algoritmos, como o DPOP e o DUCT, por exemplo, utilizam estratégias que requerem memória em ordem exponencial em relação ao tamanho do problema, com o intuito de reduzir a quantidade total de ciclos de mensagens.

Finalmente, o volume de informações compartilhadas mede a utilização do canal de comunicação entre os agentes. Estas métricas são recomendadas para medir o desempenho de um algoritmo em cenários cujo canal de comunicação não é confiável ou quando os dispositivos possuem severas limitações em relação à capacidade de armazenamento temporário de dados ou largura de banda da rede. Na plataforma FRODO, tanto o tamanho da maior mensagem enviada quanto o volume de informações compartilhadas entre os agentes são medidos em *bytes*, a partir de uma técnica de serialização de mensagens.

## 5.3 Parâmetros dos Algoritmos

Dentre os algoritmos avaliados, o COOPT, o DSA e o DUCT requerem a definição de um parâmetro em particular antes de iniciar o processo de resolução do problema. O COOPT requer a definição de uma força global de acomplamento  $K$  entre as variáveis que compartilham alguma restrição. Já o DSA utiliza uma probabilidade  $p$  antes dos agentes atribuírem novos valores para suas variáveis, cujo objetivo consiste em introduzir um processo estocástico durante a busca na tentativa de evitar máximos locais. Por fim, o DUCT realiza a exploração das soluções candidatas até que a diferença entre o custo da melhor solução e o menor *lower-bound* for menor que um dado intervalo de erro  $\epsilon$ , onde estes limites definem um intervalo de confiança como condição de parada.

Não obstante, os valores assumidos por estes parâmetros podem apresentar uma influência sobre o desempenho dos algoritmos, seja na qualidade das soluções encontradas ou no esforço computacional requerido durante a resolução do problema. Neste sentido, o uso de métodos estatísticos para avaliação do desempenho de algoritmos é uma prática amplamente adotada pela ciência da computação. Os algoritmos são, em geral, avaliados em cenários cujos parâmetros e condições iniciais são gerados aleatoriamente. Deste modo, o desempenho de cada execução do algoritmo pode ser mensurado por testes de hipótese a partir dos resultados obtidos.

### 5.3.1 Algoritmos DSA e DUCT

O trabalho de Zhang et al. (ZHANG et al., 2005) apresentou uma extensiva avaliação de desempenho do DSA em diferentes categorias de problemas. Os experimentos avaliaram o impacto do parâmetro inicial sobre a qualidade das soluções encontradas, considerando o intervalo  $0,1 \leq p \leq 1,0$ . Os resultados apresentados pelos pesquisadores revelou que o parâmetro  $p$  exerce influência apenas sobre a convergência das soluções do DSA. Além disto, os resultados também evidenciaram a probabilidade  $p = 0,7$  produziu, em média, os melhores resultados entre os cenários envolvidos. Portanto, para a avaliação experimental do COOPT, foi considerada a probabilidade  $p = 0,7$  para o DSA.

Em (OTTENS; DIMITRAKAKIS; FALTINGS, 2012), dentre várias contribuições, foi apresentada uma avaliação de desempenho do DUCT e outros algoritmos para DCOP. Neste trabalho, os pesquisadores consideraram um valor para o parâmetro  $\epsilon = 0,5$ , equivalente a um intervalo de confiança de 95%. Além disto, os pesquisadores também mensuraram o esforço computacional requerido e a qualidade das soluções encontradas, visto que o parâmetro  $\epsilon$  é utilizado como condição de parada do processo de resolução do DUCT. Desta forma, para a comparação de desempenho do COOPT, foi assumido o parâmetro  $\epsilon = 0,5$  para o DUCT.

### 5.3.2 Algoritmo COOPT

Antes de iniciar a avaliação de desempenho do COOPT, foram realizados diversos testes estatísticos para analisar a relevância do parâmetro  $K$  em relação ao custo das soluções encontradas, haja visto que este parâmetro não afeta o esforço computacional demandado em seu processo de busca, como descrito na Seção 4.2.7. Os testes estatísticos consideraram os resultados de experimentos envolvendo problemas gerados aleatoriamente envolvendo entre 10 e 50 variáveis, densidade entre 30% e 90% e topologia de redes aleatórias. Em seguida, foi realizado o teste de Friedman a partir de execuções do COOPT considerando um intervalo  $0,1 \leq K \leq 20,0$  em  $n = 50$  rounds de comunicação.

A Tabela 5.2 apresenta os resultados do teste de Friedman sobre os cenários avaliados, considerando um intervalo de confiança de 95%. Já a Figura 5.1 apresenta os resultados médios agrupados pela quantidade de variáveis envolvidas nos problemas. A partir dos resultados obtidos, é possível observar que não houve significância estatística dos valores avaliados de  $K$  sobre o custo das soluções encontradas, produzindo  $p$ -valores entre 0,0501 e 0,9736 no teste não paramétrico. Deste modo, para a avaliação de desempenho do COOPT, foi utilizada uma força de acoplamento global  $K = 1$ , a fim de considerar apenas os pesos naturais das funções de custo no processo de sincronização do algoritmo.

Tabela 5.2: Resultados do teste de Friedman para análise da relevância do parâmetro  $K$  sobre a qualidade das soluções encontradas.

(a) 10 variáveis

Restrições (% densidade)	$p$ -valor
30%	0,0501
50%	0,7797
60%	0,1604
70%	0,3998
90%	0,1363

(b) 20 variáveis

Restrições (% densidade)	$p$ -valor
30%	0,8040
50%	0,0981
60%	0,7272
70%	0,2197
90%	0,9010

(c) 30 variáveis

Restrições (% densidade)	$p$ -valor
30%	0,1138
50%	0,9736
60%	0,7014
70%	0,7485
90%	0,0611

(d) 40 variáveis

Restrições (% densidade)	$p$ -valor
30%	0,1289
50%	0,7066
60%	0,7891
70%	0,6273
90%	0,3436

(e) 50 variáveis

Restrições (% densidade)	$p$ -valor
30%	0,4973
50%	0,4563
60%	0,0668
70%	0,9061
90%	0,4447

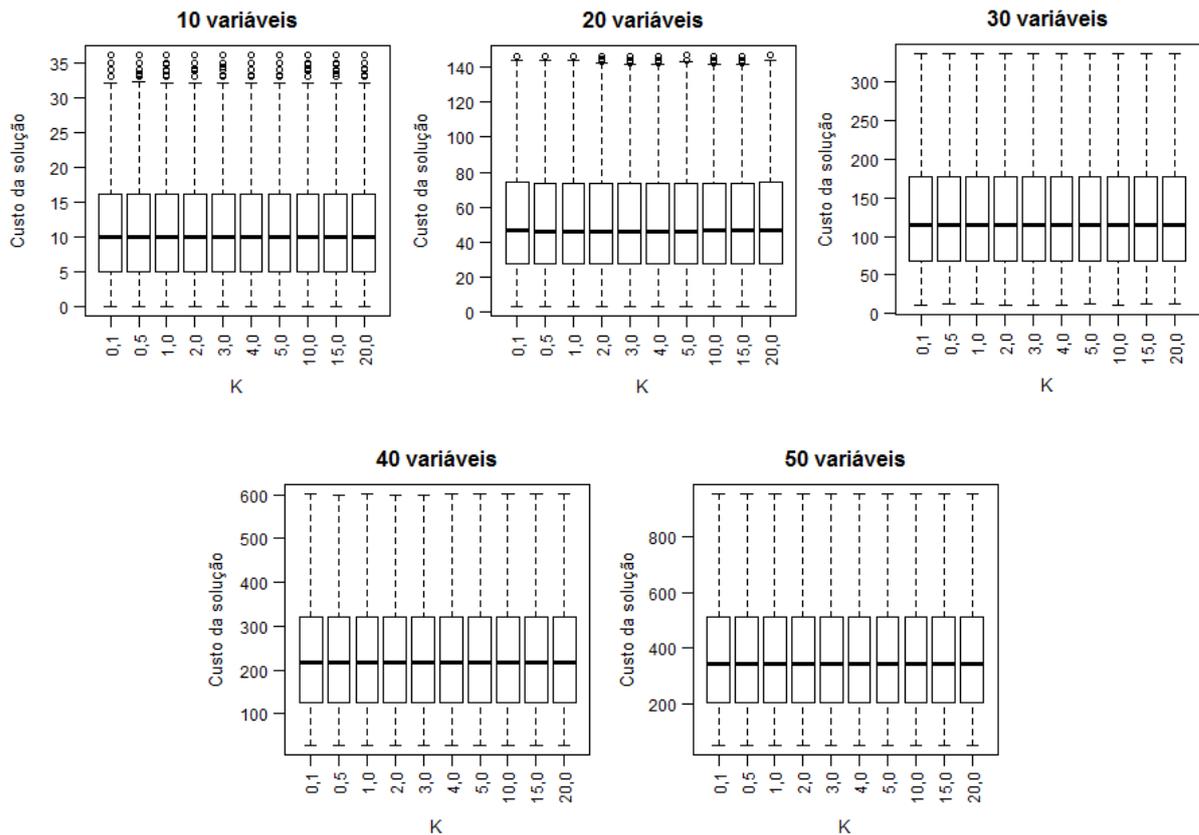


Figura 5.1: Custo médio das soluções para cada valor avaliado do parâmetro  $K$ , de acordo com o número de variáveis e restrições envolvidas nos problemas.

## 5.4 Resultados Obtidos

Conforme já mencionado, a avaliação experimental deste trabalho enfatizou o impacto do tamanho do espaço de estados dos problemas avaliados e de diferentes níveis de complexidade de rede sobre os resultados obtidos. As Tabelas 5.3, 5.4, 5.5 e 5.6 apresentam os resultados do teste de Nemenyi e valores médios obtidos para cada métrica e algoritmo utilizado nos experimentos envolvendo 10 variáveis e topologias de redes regulares, aleatórias, mundos pequenos e livres de escala, respectivamente. Por sua vez, nas Tabelas 5.7, 5.8, 5.9 e 5.10 são apresentados os resultados obtidos a partir de experimentos envolvendo 200 variáveis. Por conveniência, os demais resultados obtidos pelos experimentos (problemas envolvendo 50 e 100 variáveis) encontram-se disponíveis no apêndice deste trabalho.

Além disto, a tabulação dos resultados obtidos nos experimentos também destaca, por meio dos símbolos “●” ou “○”, os algoritmos que ocasionaram em uma melhoria ou degradação estatisticamente significativa sobre cada métrica avaliada, respectivamente. Esta classificação entre melhoria ou degradação foi realizada com base nos *rankings* gerados pelo teste de Nemenyi, considerando uma dada distância crítica (CD) entre o melhor e o pior resultado observado em cada variável envolvida no teste estatístico. A distância crítica varia conforme com o número de variáveis envolvidas no teste estatístico. Por fim, os valores médios dos resultados obtidos para cada algoritmo e métrica avaliada encontram-se destacados entre parênteses.

### 5.4.1 Qualidade das Soluções

Em geral, o COOPT alcançou as melhores soluções dos experimentos, sobretudo ao aumentar o número variáveis e os percentuais de densidade e *tightness* nos problemas. Além disto, a partir dos experimentos envolvendo 10 variáveis, foi possível observar que a qualidade das soluções encontradas pelo COOPT não apresentou uma diferença estatisticamente significativa em relação ao desempenho do DPOP, o qual é um algoritmo completo (sempre produzirá soluções ótimas).

O COOPT apresentou uma computação local eficiente em termos de qualidade das soluções encontradas, particularmente em problemas densos, isto é, em redes com um grau médio elevado. No COOPT, os agentes compartilham uma informação que representa um grau de erro local sobre uma solução parcial do problema, a fim de minimizar o custo da solução entre grupos de agentes. Não obstante, também foi observado que a qualidade das soluções encontradas pelo COOPT é ligeiramente afetada em redes esparsas.

Tabela 5.3: Resultados dos experimentos envolvendo problemas com 10 variáveis em redes regulares (CD: 2.38)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	3.45 (0.00)	4.70 (1203.98) ○	3.00 (1793.70) ●	3.00 (230.00) ●	3.00 (45529.25) ●
		DPOP	3.45 (0.00)	1.00 (128.00) ●	1.00 (79.72) ●	5.00 (524.00) ○	1.00 (19330.59) ●
		DSA	3.45 (0.00)	3.00 (500.79) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24307.07) ●
		DUCT	3.45 (0.00)	6.00 (70002.91) ○	6.00 (99301.98) ○	4.00 (417.00) ○	6.00 (4890420.04) ○
		MGM	3.75 (0.08)	2.00 (419.52) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68419.11) ○
	MGM-2	3.45 (0.00)	4.30 (1009.79) ○	5.00 (2944.57) ○	6.00 (707.32) ○	5.00 (92119.82) ○	
	50%	COOPT	3.20 (1.44) ●	4.80 (1290.40) ○	3.00 (1813.00) ●	3.00 (230.00) ●	3.00 (45971.60) ●
		DPOP	2.35 (0.00) ●	1.00 (128.00) ●	1.00 (81.70) ●	5.00 (524.00) ○	1.00 (19583.90) ●
		DSA	4.80 (4.15) ○	3.00 (501.05) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24312.96) ●
		DUCT	2.35 (0.00) ●	6.00 (63481.76) ○	6.00 (94479.94) ○	4.00 (417.00) ○	6.00 (4683274.92) ○
		MGM	4.55 (2.53) ●	2.00 (420.81) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68446.88) ○
	MGM-2	3.75 (2.40) ●	4.20 (1011.79) ○	5.00 (2969.17) ○	6.00 (757.42) ○	5.00 (96117.61) ○	
	90%	COOPT	3.20 (96.51) ●	4.70 (1232.00) ○	3.00 (1803.20) ●	3.00 (230.00) ●	3.00 (45785.44) ●
		DPOP	1.30 (71.70) ○	1.00 (128.00) ●	1.00 (79.80) ●	5.00 (524.00) ○	1.00 (19410.00) ●
		DSA	5.55 (119.76) ○	3.00 (500.03) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24311.68) ●
DUCT		1.80 (73.49) ●	6.00 (48781.68) ○	6.00 (76408.30) ○	4.00 (417.00) ○	6.00 (3778807.87) ○	
MGM		4.45 (112.60) ○	2.00 (420.45) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68468.32) ○	
MGM-2	4.70 (113.70) ○	4.30 (996.68) ○	5.00 (3105.12) ○	6.00 (752.30) ○	5.00 (107658.65) ○		
50%	30%	COOPT	3.60 (2.47) ●	4.00 (2275.20) ○	3.00 (2842.00) ●	3.00 (230.00) ●	3.00 (75667.36) ●
		DPOP	1.80 (0.00) ●	5.00 (26625.40) ○	1.00 (119.50) ●	6.00 (13302.70) ○	1.70 (58039.80) ●
		DSA	4.40 (4.72) ○	2.00 (997.45) ●	2.00 (2000.00) ●	2.00 (228.00) ●	1.30 (48660.80) ●
		DUCT	1.80 (0.00) ●	6.00 (219432.32) ○	6.00 (159059.22) ○	4.00 (417.00) ○	6.00 (9640318.16) ○
		MGM	5.10 (5.82) ○	1.00 (825.61) ●	4.00 (4080.00) ○	1.00 (222.00) ●	4.00 (136895.36) ○
	MGM-2	4.30 (4.38) ○	3.00 (2016.19) ●	5.00 (5910.96) ○	5.00 (761.92) ○	5.00 (187137.36) ○	
	50%	COOPT	2.70 (38.85) ●	4.00 (2519.20) ○	3.00 (2842.00) ●	3.00 (230.00) ●	2.60 (75727.20) ●
		DPOP	1.15 (17.50) ●	5.00 (91751.40) ○	1.00 (119.90) ●	6.00 (31838.50) ○	2.60 (92124.30) ●
		DSA	4.80 (61.58) ○	2.00 (996.39) ●	2.00 (2000.00) ●	2.00 (228.00) ●	1.00 (48651.20) ●
		DUCT	2.45 (28.80) ●	6.00 (212090.72) ○	6.00 (149950.64) ○	4.00 (417.00) ○	6.00 (9285882.30) ○
		MGM	4.90 (62.84) ○	1.00 (824.29) ●	4.00 (4080.00) ○	1.00 (222.00) ●	3.80 (136940.16) ○
	MGM-2	5.00 (63.99) ○	3.00 (2001.14) ●	5.00 (5985.18) ○	5.00 (762.42) ○	5.00 (195676.50) ○	
	90%	COOPT	2.40 (321.54) ●	4.00 (2421.60) ○	3.00 (2842.00) ●	3.00 (230.00) ●	2.90 (75739.04) ●
		DPOP	1.25 (308.60) ●	5.00 (33588.50) ○	1.00 (119.20) ●	6.00 (14539.80) ○	1.90 (62697.50) ●
		DSA	4.50 (353.02) ○	2.00 (996.88) ●	2.00 (2000.00) ●	2.00 (228.00) ●	1.20 (48656.96) ●
DUCT		2.55 (323.62) ○	6.00 (194050.24) ○	6.00 (141215.46) ○	4.00 (417.00) ○	6.00 (8585391.26) ○	
MGM		5.20 (367.05) ○	1.00 (825.39) ●	4.00 (4080.00) ○	1.00 (222.00) ●	4.00 (136962.88) ○	
MGM-2	5.10 (366.08) ○	3.00 (1978.35) ●	5.00 (6019.03) ○	5.00 (753.00) ○	5.00 (198561.18) ○		
90%	30%	COOPT	3.20 (54.22) ●	4.00 (4612.00) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133278.88) ●
		DPOP	1.00 (25.20) ●	6.00 (2097152.00) ○	1.00 (196.00) ●	6.00 (264020.00) ○	5.00 (670252.00) ○
		DSA	4.70 (74.40) ○	2.00 (1969.34) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97393.92) ●
		DUCT	2.00 (40.29) ●	5.00 (504534.08) ○	6.00 (170045.46) ○	4.00 (417.00) ○	6.00 (11038058.97) ○
		MGM	5.40 (90.18) ○	1.00 (1632.57) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.00 (273864.96) ●
	MGM-2	4.70 (72.52) ○	3.00 (4000.94) ●	5.00 (11832.38) ○	5.00 (765.84) ○	4.00 (375709.86) ○	
	50%	COOPT	2.10 (253.33) ●	4.00 (4612.00) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133456.80) ●
		DPOP	1.00 (233.80) ●	6.00 (2097152.00) ○	1.00 (196.20) ●	6.00 (264020.00) ○	5.00 (670261.60) ○
		DSA	4.10 (291.50) ○	2.00 (1969.30) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97421.44) ●
		DUCT	3.10 (265.30) ●	5.00 (483922.88) ○	6.00 (165365.10) ○	4.00 (417.00) ○	6.00 (10718535.60) ○
		MGM	5.10 (308.80) ○	1.00 (1632.47) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.00 (273918.72) ●
	MGM-2	5.60 (323.50) ○	3.00 (3969.41) ●	5.00 (11861.73) ○	5.00 (754.44) ○	4.00 (379785.83) ○	
	90%	COOPT	1.90 (1010.30) ●	4.00 (4612.00) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133304.48) ●
		DPOP	1.20 (995.20) ●	6.00 (2097152.00) ○	1.00 (196.30) ●	6.00 (264020.00) ○	5.00 (670261.20) ○
		DSA	5.40 (1094.60) ○	2.00 (1970.07) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97410.56) ●
DUCT		3.00 (1045.31) ●	5.00 (482456.32) ○	6.00 (166076.80) ○	4.00 (417.00) ○	6.00 (10757585.20) ○	
MGM		4.80 (1082.95) ○	1.00 (1632.37) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.00 (273911.68) ●	
MGM-2	4.70 (1084.27) ○	3.00 (3942.93) ●	5.00 (11882.18) ○	5.00 (770.96) ○	4.00 (381621.93) ○		

Tabela 5.4: Resultados dos experimentos envolvendo problemas com 10 variáveis em redes aleatórias (CD: 2.38)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	2.65 (0.10)	3.70 (1361.90) ◦	3.00 (2116.80) ●	3.00 (230.00) ●	3.00 (54755.44) ●
		DPOP	2.50 (0.00)	1.00 (435.90) ●	1.00 (86.70) ●	5.20 (654.80) ◦	1.00 (22972.30) ●
		DSA	4.45 (2.61)	3.20 (1181.70) ●	2.00 (1300.00) ●	2.00 (228.00) ●	2.00 (31620.40) ●
		DUCT	2.50 (0.00)	6.00 (69741.30) ◦	6.00 (92176.74) ◦	4.00 (417.00) ◦	6.00 (4464381.47) ◦
		MGM	4.55 (1.64)	2.10 (979.98) ●	4.00 (2652.00) ◦	1.00 (222.00) ●	4.00 (88958.24) ◦
	MGM-2	4.35 (1.05)	5.00 (2072.09) ◦	5.00 (3836.95) ◦	5.00 (3881.35) ◦	5.00 (121390.40) ◦	
	50%	COOPT	2.40 (1.99) ●	3.70 (1411.00) ◦	3.00 (2116.80) ●	3.00 (230.00) ●	3.00 (54755.84) ●
		DPOP	1.80 (1.20) ●	1.00 (551.00) ●	1.00 (87.30) ●	5.50 (676.60) ◦	1.00 (23169.40) ●
		DSA	4.90 (22.21) ◦	3.20 (1182.24) ●	2.00 (1300.00) ●	2.00 (228.00) ●	2.00 (31615.84) ●
		DUCT	1.80 (1.20) ●	6.00 (63605.52) ◦	6.00 (79564.16) ◦	4.00 (417.00) ◦	6.00 (3869873.70) ◦
		MGM	5.10 (24.48) ◦	2.10 (981.81) ●	4.00 (2652.00) ◦	1.00 (222.00) ●	4.00 (88999.76) ◦
	MGM-2	5.00 (21.55) ◦	5.00 (2088.56) ◦	5.00 (3881.35) ◦	5.50 (742.82) ◦	5.00 (126143.65) ◦	
	90%	COOPT	2.30 (169.85) ●	4.00 (1484.90) ◦	3.00 (2146.20) ●	3.00 (230.00) ●	3.00 (55410.32) ●
		DPOP	1.35 (171.90) ◦	1.00 (448.80) ●	1.00 (88.10) ●	5.00 (676.60) ◦	1.00 (23448.60) ●
		DSA	4.70 (211.26) ◦	3.10 (1156.74) ●	2.00 (1300.00) ●	2.00 (228.00) ●	2.00 (31619.52) ●
DUCT		2.35 (175.54) ●	6.00 (58038.11) ◦	6.00 (84675.64) ◦	4.00 (417.00) ◦	6.00 (4195102.40) ◦	
MGM		5.70 (226.32) ◦	2.00 (959.36) ●	4.00 (2652.00) ◦	1.00 (222.00) ●	4.00 (89011.60) ◦	
MGM-2	4.60 (211.50) ◦	4.90 (1949.23) ◦	5.00 (3991.36) ◦	6.00 (766.54) ◦	5.00 (135957.11) ◦		
50%	30%	COOPT	3.00 (5.86) ●	3.10 (2388.70) ●	3.00 (3038.00) ●	3.00 (230.00) ●	3.00 (81433.44) ●
		DPOP	1.85 (0.00) ●	5.00 (10650.10) ◦	1.00 (124.70) ●	6.00 (4014.10) ◦	1.10 (42185.60) ●
		DSA	4.70 (9.94) ◦	2.00 (1551.14) ●	2.00 (2200.00) ●	2.00 (228.00) ●	1.90 (53537.36) ●
		DUCT	1.85 (0.00) ●	6.00 (150588.89) ◦	6.00 (130177.94) ◦	4.00 (417.00) ◦	6.00 (7393402.87) ◦
		MGM	5.25 (13.49) ◦	1.00 (1288.48) ●	4.00 (4488.00) ◦	1.00 (222.00) ●	4.00 (150595.20) ◦
	MGM-2	4.35 (10.25) ◦	3.90 (2826.21) ◦	5.00 (6500.76) ◦	5.00 (751.79) ◦	5.00 (205668.30) ◦	
	50%	COOPT	2.80 (41.37) ●	3.20 (2455.40) ●	3.00 (3038.00) ●	3.00 (230.00) ●	3.00 (81509.52) ●
		DPOP	1.15 (32.80) ●	5.00 (9831.60) ◦	1.00 (124.30) ●	6.00 (3710.00) ◦	1.00 (40960.40) ●
		DSA	5.00 (74.00) ◦	2.00 (1575.10) ●	2.00 (2200.00) ●	2.00 (228.00) ●	2.00 (53546.80) ●
		DUCT	2.05 (37.79) ●	6.00 (132894.51) ◦	6.00 (127001.32) ◦	4.00 (417.00) ◦	6.00 (7219479.49) ◦
		MGM	4.85 (74.68) ◦	1.00 (1307.93) ●	4.00 (4488.00) ◦	1.00 (222.00) ●	4.00 (150631.44) ◦
	MGM-2	5.15 (79.32) ◦	3.80 (2791.75) ◦	5.00 (6576.72) ◦	5.00 (751.24) ◦	5.00 (213768.04) ◦	
	90%	COOPT	2.50 (401.85) ●	3.10 (2379.20) ●	3.00 (3038.00) ●	3.00 (230.00) ●	3.00 (81521.52) ●
		DPOP	1.45 (385.30) ●	4.90 (6349.90) ◦	1.00 (124.30) ●	6.00 (2777.00) ◦	1.00 (39169.10) ●
		DSA	4.50 (458.79) ◦	2.00 (1623.95) ●	2.00 (2200.00) ●	2.00 (228.00) ●	2.00 (53535.20) ●
DUCT		2.05 (391.47) ●	6.00 (128400.64) ◦	6.00 (115122.64) ◦	4.00 (417.00) ◦	6.00 (6444523.23) ◦	
MGM		5.25 (459.01) ◦	1.00 (1347.16) ●	4.00 (4488.00) ◦	1.00 (222.00) ●	4.00 (150646.64) ◦	
MGM-2	5.25 (463.48) ◦	4.00 (2822.44) ◦	5.00 (6625.61) ◦	5.00 (762.22) ◦	5.00 (218724.54) ◦		
90%	30%	COOPT	2.50 (35.28) ●	3.90 (4433.90) ◦	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133290.08) ●
		DPOP	1.10 (21.00) ●	6.00 (930611.80) ◦	1.00 (196.60) ●	6.00 (145465.40) ◦	4.00 (324625.50) ◦
		DSA	4.40 (64.16) ◦	2.00 (2214.40) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97389.28) ●
		DUCT	2.50 (33.66) ●	5.00 (407854.94) ◦	6.00 (165809.70) ◦	4.00 (417.00) ◦	6.00 (10600234.96) ◦
		MGM	5.60 (75.85) ◦	1.00 (1836.44) ●	4.00 (8160.00) ◦	1.00 (222.00) ●	3.40 (273896.64) ◦
	MGM-2	4.90 (67.74) ◦	3.10 (4156.55) ●	5.00 (11829.75) ◦	5.00 (765.18) ◦	4.60 (375112.47) ◦	
	50%	COOPT	2.60 (265.24) ●	4.00 (4399.70) ◦	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133249.52) ●
		DPOP	1.15 (244.00) ●	6.00 (681574.90) ◦	1.00 (196.10) ●	6.00 (105947.20) ◦	3.60 (280118.80) ●
		DSA	4.90 (314.84) ◦	2.00 (2214.36) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97390.24) ●
		DUCT	2.65 (264.78) ●	5.00 (378825.32) ◦	6.00 (158066.46) ◦	4.00 (417.00) ◦	6.00 (10068233.14) ◦
		MGM	4.60 (314.20) ◦	1.00 (1836.56) ●	4.00 (8160.00) ◦	1.00 (222.00) ●	3.60 (273930.48) ◦
	MGM-2	5.10 (330.61) ◦	3.00 (4151.44) ●	5.00 (11856.59) ◦	5.00 (771.98) ◦	4.80 (378923.60) ◦	
	90%	COOPT	2.25 (976.51) ●	4.00 (4404.60) ◦	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133325.68) ●
		DPOP	1.25 (957.20) ●	6.00 (871629.20) ◦	1.00 (196.30) ●	6.00 (125706.30) ◦	3.70 (293693.40) ◦
		DSA	4.50 (1032.22) ◦	2.00 (2214.33) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97355.68) ●
DUCT		2.70 (980.43) ●	5.00 (376862.14) ◦	6.00 (156779.06) ◦	4.00 (417.00) ◦	6.00 (9975077.77) ◦	
MGM		5.20 (1048.21) ◦	1.00 (1836.47) ●	4.00 (8160.00) ◦	1.00 (222.00) ●	3.60 (273961.44) ◦	
MGM-2	5.10 (1059.15) ◦	3.00 (4134.20) ●	5.00 (11880.07) ◦	5.00 (763.98) ◦	4.70 (381629.60) ◦		

Tabela 5.5: Resultados dos experimentos envolvendo problemas com 10 variáveis em redes de mundos pequenos (CD: 2.38)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	3.70 (22.50)	4.00 (1044.50) ○	3.00 (1832.60) ●	3.00 (230.00) ●	3.00 (46356.40) ●
		DPOP	3.05 (0.00)	1.00 (166.60) ●	1.00 (77.80) ●	5.10 (545.80) ○	1.00 (19254.50) ●
		DSA	3.60 (0.29)	3.00 (813.63) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24306.64) ●
		DUCT	3.05 (0.00)	6.00 (57999.38) ○	6.00 (84256.86) ○	4.00 (417.00) ○	6.00 (3875335.43) ○
		MGM	4.30 (1.24)	2.00 (675.65) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68418.96) ○
	MGM-2	3.30 (0.18)	5.00 (1438.55) ○	5.00 (2946.42) ○	5.00 (2946.42) ○	5.90 (711.10) ○	5.00 (92531.57) ○
	50%	COOPT	2.60 (15.53) ●	3.90 (977.90) ○	3.00 (1842.40) ●	3.00 (230.00) ●	3.00 (46586.80) ●
		DPOP	1.85 (0.00) ●	1.00 (128.00) ●	1.00 (78.70) ●	5.00 (524.00) ○	1.00 (19283.63) ●
		DSA	4.55 (5.54) ○	3.10 (788.49) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24316.96) ●
		DUCT	1.85 (0.00) ●	6.00 (47599.72) ○	6.00 (77129.20) ○	4.00 (417.00) ○	6.00 (3496894.34) ○
		MGM	5.05 (6.83) ○	2.00 (654.66) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68449.92) ○
	MGM-2	5.10 (6.73) ○	5.00 (1415.74) ○	5.00 (2990.16) ○	6.00 (742.84) ○	5.00 (97690.04) ○	
	90%	COOPT	3.00 (84.34) ●	4.00 (1112.60) ○	3.00 (1842.40) ●	3.00 (230.00) ●	3.00 (46573.84) ●
		DPOP	1.60 (79.70) ○	1.00 (128.10) ●	1.00 (79.60) ●	5.00 (524.00) ○	1.00 (19418.10) ●
		DSA	5.10 (131.74) ○	3.10 (763.45) ●	2.00 (1000.00) ●	2.00 (228.00) ●	2.00 (24316.96) ●
DUCT		1.80 (81.07) ●	6.00 (44215.36) ○	6.00 (66298.32) ○	4.00 (417.00) ○	6.00 (3084588.15) ○	
MGM		4.70 (126.72) ○	2.10 (633.41) ●	4.00 (2040.00) ○	1.00 (222.00) ●	4.00 (68472.56) ○	
MGM-2	4.80 (129.69) ○	4.80 (1327.21) ○	5.00 (3126.49) ○	6.00 (758.94) ○	5.00 (109650.02) ○		
50%	30%	COOPT	3.55 (7.93) ●	3.60 (2333.30)	3.00 (2842.00) ●	3.00 (230.00) ●	3.00 (75669.12) ●
		DPOP	1.90 (0.00) ●	4.80 (3687.60) ○	1.00 (116.60) ●	6.00 (2155.00) ○	1.00 (36764.20) ●
		DSA	4.90 (8.37) ○	2.00 (1354.13) ●	2.00 (2000.00) ●	2.00 (228.00) ●	2.00 (48663.12) ●
		DUCT	1.90 (0.00) ●	6.00 (155839.74) ○	6.00 (140100.94) ○	4.00 (417.00) ○	6.00 (7940211.76) ○
		MGM	4.45 (5.39) ○	1.00 (1125.96) ●	4.00 (4080.00) ○	1.00 (222.00) ●	4.00 (136885.84) ○
	MGM-2	4.30 (4.52) ○	3.60 (2451.68)	5.00 (5913.14) ○	5.00 (755.12) ○	5.00 (187622.53) ○	
	50%	COOPT	2.80 (25.32) ●	3.10 (2201.60) ●	3.00 (2842.00) ●	3.00 (230.00) ●	3.00 (75727.76) ●
		DPOP	1.25 (15.80) ●	5.00 (7271.40) ○	1.00 (117.90) ●	6.00 (3703.10) ○	1.10 (38337.70) ●
		DSA	4.90 (59.20) ○	2.00 (1329.69) ●	2.00 (2000.00) ●	2.00 (228.00) ●	1.90 (48666.80) ●
		DUCT	2.35 (23.48) ●	6.00 (133193.06) ○	6.00 (126944.44) ○	4.00 (417.00) ○	6.00 (7132700.20) ○
		MGM	5.30 (62.28) ○	1.00 (1105.20) ●	4.00 (4080.00) ○	1.00 (222.00) ●	4.00 (136934.88) ○
	MGM-2	4.40 (58.63) ○	3.90 (2400.41) ○	5.00 (5976.26) ○	5.00 (749.90) ○	5.00 (194539.30) ○	
	90%	COOPT	2.30 (306.12) ●	3.20 (2228.90) ●	3.00 (2842.00) ●	3.00 (230.00) ●	3.00 (75696.72) ●
		DPOP	1.20 (293.60) ●	4.90 (9728.80) ○	1.00 (117.10) ●	6.00 (3703.10) ○	1.10 (38609.70) ●
		DSA	4.50 (348.02) ○	2.00 (1379.17) ●	2.00 (2000.00) ●	2.00 (228.00) ●	1.90 (48679.76) ●
DUCT		2.50 (304.79) ●	6.00 (124456.78) ○	6.00 (115458.90) ○	4.00 (417.00) ○	6.00 (6530142.84) ○	
MGM		5.40 (366.33) ○	1.00 (1147.52) ●	4.00 (4080.00) ○	1.00 (222.00) ●	4.00 (136972.40) ○	
MGM-2	5.10 (356.39) ○	3.90 (2434.11) ○	5.00 (6021.61) ○	5.00 (754.58) ○	5.00 (198265.17) ○		
90%	30%	COOPT	3.25 (41.76) ●	3.80 (4458.30) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133196.48) ●
		DPOP	1.05 (17.30) ●	6.00 (1205862.90) ○	1.00 (196.50) ●	6.00 (184983.60) ○	4.40 (393786.90) ○
		DSA	4.70 (65.15) ○	2.00 (2214.42) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97401.68) ●
		DUCT	2.40 (29.63) ●	5.00 (414989.30) ○	6.00 (165277.72) ○	4.00 (417.00) ○	6.00 (10580019.36) ○
		MGM	4.80 (68.94) ○	1.00 (1836.47) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.20 (273860.40) ●
	MGM-2	4.80 (67.53) ○	3.20 (4207.38) ●	5.00 (11830.75) ○	5.00 (755.72) ○	4.40 (375193.80) ○	
	50%	COOPT	2.95 (246.17) ●	4.00 (4463.20) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133372.80) ●
		DPOP	1.05 (204.30) ●	6.00 (1068237.40) ○	1.00 (196.20) ●	6.00 (165224.50) ○	4.40 (359188.00) ○
		DSA	4.40 (269.85) ○	2.00 (2214.28) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97374.72) ●
		DUCT	2.50 (229.92) ●	5.00 (391540.02) ○	6.00 (157660.90) ○	4.00 (417.00) ○	6.00 (10085542.87) ○
		MGM	5.00 (290.98) ○	1.00 (1836.55) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.10 (273939.28) ●
	MGM-2	5.10 (290.47) ○	3.00 (4098.86) ●	5.00 (11855.64) ○	5.00 (772.22) ○	4.50 (378626.94) ○	
	90%	COOPT	1.70 (964.47) ●	4.00 (4570.70) ○	3.00 (4802.00) ●	3.00 (230.00) ●	2.00 (133268.32) ●
		DPOP	1.30 (956.00) ●	6.00 (1291059.90) ○	1.00 (196.30) ●	6.00 (204742.70) ○	4.70 (413549.30) ○
		DSA	4.40 (1033.69) ○	2.00 (2214.31) ●	2.00 (4000.00) ●	2.00 (228.00) ●	1.00 (97395.20) ●
DUCT		3.20 (993.42) ●	5.00 (407293.80) ○	6.00 (160590.56) ○	4.00 (417.00) ○	6.00 (10362265.80) ○	
MGM		5.20 (1055.90) ○	1.00 (1836.47) ●	4.00 (8160.00) ○	1.00 (222.00) ●	3.00 (273955.60) ○	
MGM-2	5.20 (1045.64) ○	3.00 (4100.07) ●	5.00 (11874.06) ○	5.00 (762.22) ○	4.30 (380608.35) ○		

Tabela 5.6: Resultados dos experimentos envolvendo problemas com 10 variáveis em redes livres de escala (CD: 2.38)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	4.10 (3.49)	2.20 (845.30) ●	3.00 (1764.00) ●	3.00 (230.00) ●	3.00 (44102.72) ●
		DPOP	2.95 (0.00)	1.00 (18.10) ●	1.00 (73.00) ●	5.00 (453.00) ○	1.00 (17872.90) ●
		DSA	3.35 (0.77)	3.90 (1304.25) ○	2.00 (900.00) ●	2.00 (228.00) ●	2.00 (21877.92) ●
		DUCT	2.95 (0.00)	6.00 (19458.70) ○	6.00 (59292.98) ○	4.00 (417.00) ○	6.00 (2359879.94) ○
		MGM	4.45 (1.71)	2.90 (1081.67) ●	4.00 (1836.00) ○	1.00 (222.00) ●	4.00 (61580.72) ○
	MGM-2	3.20 (0.22)	5.00 (2235.50) ○	5.00 (2650.56) ○	6.00 (713.92) ○	5.00 (82959.84) ○	
	50%	COOPT	3.20 (4.16) ●	2.60 (855.30) ●	3.00 (1764.00) ●	3.00 (230.00) ●	3.00 (44112.80) ●
		DPOP	2.10 (0.20) ●	1.00 (17.70) ●	1.00 (73.30) ●	5.00 (453.00) ○	1.00 (17886.10) ●
		DSA	5.00 (10.10) ○	3.70 (1304.80) ○	2.00 (900.00) ●	2.00 (228.00) ●	2.00 (21874.32) ●
		DUCT	2.10 (0.20) ●	6.00 (18496.02) ○	6.00 (53605.00) ○	4.00 (417.00) ○	6.00 (2137983.70) ○
		MGM	5.10 (7.76) ○	2.70 (1082.46) ●	4.00 (1836.00) ○	1.00 (222.00) ●	4.00 (61594.16) ○
	MGM-2	3.50 (5.19) ●	5.00 (2312.77) ○	5.00 (2675.33) ○	6.00 (741.11) ○	5.00 (86335.44) ○	
90%	COOPT	3.60 (92.19) ●	2.10 (836.80) ●	3.00 (1764.00) ●	3.00 (230.00) ●	3.00 (44119.68) ●	
	DPOP	1.50 (67.40) ●	1.00 (18.10) ●	1.00 (72.50) ●	5.00 (453.00) ○	1.00 (17848.40) ●	
	DSA	5.20 (109.92) ○	4.00 (1205.71) ○	2.00 (900.00) ●	2.00 (228.00) ●	2.00 (21874.32) ●	
	DUCT	1.50 (67.40) ●	6.00 (15993.97) ○	6.00 (50647.62) ○	4.00 (417.00) ○	6.00 (2030995.62) ○	
	MGM	4.20 (103.82) ○	2.90 (999.95) ●	4.00 (1836.00) ○	1.00 (222.00) ●	4.00 (61615.36) ○	
MGM-2	5.00 (109.28) ○	5.00 (2158.64) ○	5.00 (2867.40) ○	6.00 (756.40) ○	5.00 (103632.43) ○		
50%	30%	COOPT	3.75 (8.64) ●	2.90 (1726.80) ●	3.00 (2548.00) ●	3.00 (230.00) ●	3.00 (67051.92) ●
		DPOP	2.05 (0.00) ●	1.00 (449.70) ●	1.00 (104.10) ●	5.10 (633.00) ○	1.00 (28340.30) ●
		DSA	4.70 (6.75) ○	3.70 (1869.75) ○	2.00 (1700.00) ●	2.00 (228.00) ●	2.00 (41335.12) ●
		DUCT	2.05 (0.00) ●	6.00 (68332.58) ○	6.00 (98960.68) ○	4.00 (417.00) ○	6.00 (4974661.11) ○
		MGM	4.50 (5.77) ○	2.40 (1550.53) ●	4.00 (3468.00) ○	1.00 (222.00) ●	4.00 (116345.84) ○
	MGM-2	3.95 (5.36) ●	5.00 (3208.96) ○	5.00 (5025.64) ○	5.90 (767.94) ○	5.00 (159198.34) ○	
	50%	COOPT	2.90 (26.22) ●	2.70 (1629.70) ●	3.00 (2548.00) ●	3.00 (230.00) ●	3.00 (67096.40) ●
		DPOP	1.60 (11.00) ●	1.00 (398.60) ●	1.00 (104.10) ●	5.10 (633.00) ○	1.00 (28282.10) ●
		DSA	4.80 (47.74) ○	3.80 (1869.94) ○	2.00 (1700.00) ●	2.00 (228.00) ●	2.00 (41351.76) ●
		DUCT	1.70 (11.12) ●	6.00 (59382.17) ○	6.00 (84429.70) ○	4.00 (417.00) ○	6.00 (4221396.87) ○
		MGM	5.30 (54.24) ○	2.50 (1550.92) ●	4.00 (3468.00) ○	1.00 (222.00) ●	4.00 (116385.84) ○
	MGM-2	4.70 (47.95) ○	5.00 (3215.74) ○	5.00 (5085.83) ○	5.90 (753.58) ○	5.00 (165858.31) ○	
	90%	COOPT	2.85 (261.51) ●	2.90 (1602.20) ●	3.00 (2548.00) ●	3.00 (230.00) ●	3.00 (67129.04) ●
		DPOP	1.60 (244.70) ●	1.30 (590.30) ●	1.00 (104.00) ●	5.30 (733.90) ○	1.00 (28457.20) ●
		DSA	4.80 (303.62) ○	3.50 (1771.73) ●	2.00 (1700.00) ●	2.00 (228.00) ●	2.00 (41340.80) ●
DUCT		1.75 (244.84) ○	6.00 (53336.16) ○	6.00 (77908.00) ○	4.00 (417.00) ○	6.00 (3910675.50) ○	
MGM		4.80 (306.74) ○	2.30 (1469.37) ●	4.00 (3468.00) ○	1.00 (222.00) ●	4.00 (116413.60) ○	
MGM-2	5.20 (324.55) ○	5.00 (3022.42) ○	5.00 (5195.97) ○	5.70 (759.62) ○	5.00 (175383.94) ○		
90%	30%	COOPT	3.35 (24.26) ●	3.00 (2998.80) ●	3.00 (3822.00) ●	3.00 (230.00) ●	3.00 (104417.68) ●
		DPOP	1.50 (6.00) ○	5.00 (11470.20) ○	1.00 (156.20) ●	6.00 (3088.00) ○	1.00 (53333.70) ●
		DSA	4.50 (30.70) ○	2.00 (2165.20) ●	2.00 (3000.00) ●	2.00 (228.00) ●	2.00 (73021.28) ●
		DUCT	1.90 (7.90) ●	6.00 (179257.34) ○	6.00 (135690.02) ○	4.00 (417.00) ○	6.00 (7747182.21) ○
		MGM	5.90 (39.77) ○	1.00 (1795.56) ●	4.00 (6120.00) ○	1.00 (222.00) ●	4.00 (205362.32) ○
	MGM-2	3.85 (27.32) ●	4.00 (3826.30) ○	5.00 (8876.92) ○	5.00 (772.90) ○	5.00 (281811.45) ○	
	50%	COOPT	3.10 (121.62) ●	3.00 (3067.20) ●	3.00 (3822.00) ●	3.00 (230.00) ●	3.00 (104508.72) ●
		DPOP	1.25 (90.40) ●	5.00 (20890.50) ○	1.00 (156.10) ●	6.00 (5258.10) ○	1.10 (58303.60) ●
		DSA	4.50 (151.88) ○	2.00 (2165.19) ●	2.00 (3000.00) ●	2.00 (228.00) ●	1.90 (73002.72) ●
		DUCT	1.85 (92.79) ●	6.00 (187265.22) ○	6.00 (121847.84) ○	4.00 (417.00) ○	6.00 (7013234.45) ○
		MGM	5.10 (163.17) ○	1.00 (1795.71) ●	4.00 (6120.00) ○	1.00 (222.00) ●	4.00 (205428.08) ○
	MGM-2	5.20 (163.35) ○	4.00 (3820.01) ○	5.00 (8928.66) ○	5.00 (769.80) ○	5.00 (287397.65) ○	
	90%	COOPT	2.85 (678.37) ●	3.00 (2958.00) ●	3.00 (3822.00) ●	3.00 (230.00) ●	3.00 (104492.00) ●
		DPOP	1.60 (653.00) ●	5.00 (17614.00) ○	1.00 (156.10) ●	6.00 (5258.10) ○	1.00 (54893.50) ●
		DSA	4.35 (716.44) ○	2.00 (2189.92) ●	2.00 (3000.00) ●	2.00 (228.00) ●	2.00 (72992.64) ●
DUCT		1.85 (656.15) ●	6.00 (165665.58) ○	6.00 (120472.66) ○	4.00 (417.00) ○	6.00 (6867418.71) ○	
MGM		5.50 (753.12) ○	1.00 (1815.77) ●	4.00 (6120.00) ○	1.00 (222.00) ●	4.00 (205438.40) ○	
MGM-2	4.85 (730.13) ○	4.00 (3837.11) ○	5.00 (8944.13) ○	5.00 (778.16) ○	5.00 (289361.62) ○		

Tabela 5.7: Resultados dos experimentos envolvendo problemas com 200 variáveis em redes regulares (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (49956.65) ●	4.00 (580451.14) ○	2.00 (597702.00) ●	2.50 (232.00)	2.00 (17420458.16) ●
		DSA	2.00 (51081.79) ●	2.00 (15047.61) ●	1.00 (590000.00) ●	2.50 (232.00)	1.00 (14443553.40) ●
		MGM	3.00 (52810.96) ○	1.00 (12233.13) ●	3.00 (1203600.00) ○	1.00 (226.00) ●	3.00 (40442960.04) ○
		MGM-2	4.00 (53158.63) ○	3.00 (31023.90) ○	4.00 (1737639.03) ○	4.00 (817.00) ○	4.00 (54135341.32) ○
	50%	COOPT	1.00 (101263.08) ●	4.00 (579695.40) ○	2.00 (597702.00) ●	2.50 (232.00)	2.00 (17411810.16) ●
		DSA	2.00 (102669.83) ●	2.00 (15024.25) ●	1.00 (590000.00) ●	2.50 (232.00)	1.00 (14444211.85) ●
		MGM	3.10 (104671.28) ○	1.00 (12229.29) ●	3.00 (1203600.00) ○	1.00 (226.00) ●	3.00 (40442881.04) ○
		MGM-2	3.90 (105374.52) ○	3.00 (30870.86) ○	4.00 (1737798.24) ○	4.00 (817.00) ○	4.00 (54185869.30) ○
	90%	COOPT	1.00 (219419.76) ●	4.00 (578225.90) ○	2.00 (597702.00) ●	2.50 (232.00)	2.00 (17398952.88) ●
		DSA	2.00 (221029.18) ●	2.00 (15031.21) ●	1.00 (590000.00) ●	2.50 (232.00)	1.00 (14444712.76) ●
		MGM	3.10 (223275.29) ○	1.00 (12227.07) ●	3.00 (1203600.00) ○	1.00 (226.00) ●	3.00 (40443919.44) ○
		MGM-2	3.90 (223530.24) ○	3.00 (30890.78) ○	4.00 (1737807.26) ○	4.00 (817.00) ○	4.00 (54182924.77) ○
50%	30%	COOPT	1.00 (97629.16) ●	4.00 (978558.90) ○	2.00 (999702.00) ●	2.50 (232.00)	2.00 (28950435.92) ●
		DSA	2.00 (98945.08) ●	2.00 (24954.26) ●	1.00 (980000.00) ●	2.50 (232.00)	1.00 (24238500.66) ●
		MGM	3.20 (106298.15) ○	1.00 (20320.96) ●	3.00 (2019600.00) ○	1.00 (226.00) ●	3.00 (67827363.12) ○
		MGM-2	3.80 (106725.96) ○	3.00 (52644.90) ○	4.00 (2914876.32) ○	4.00 (817.00) ○	4.00 (90590911.43) ○
	50%	COOPT	1.00 (186733.44) ●	4.00 (975619.50) ○	2.00 (999702.00) ●	2.50 (232.00)	2.00 (28941739.76) ●
		DSA	2.00 (188507.37) ●	2.00 (24956.36) ●	1.00 (980000.00) ●	2.50 (232.00)	1.00 (24239213.46) ●
		MGM	3.40 (197556.74) ○	1.00 (20324.56) ●	3.00 (2019600.00) ○	1.00 (226.00) ●	3.00 (67827109.68) ○
		MGM-2	3.60 (198112.26) ○	3.00 (52533.08) ○	4.00 (2914911.57) ○	4.00 (817.00) ○	4.00 (90604644.82) ○
	90%	COOPT	1.00 (384261.64) ●	4.00 (979538.70) ○	2.00 (999702.00) ●	2.50 (232.00)	2.00 (28918027.28) ●
		DSA	2.00 (386099.05) ●	2.00 (24932.48) ●	1.00 (980000.00) ●	2.50 (232.00)	1.00 (24241007.34) ●
		MGM	3.10 (395247.72) ○	1.00 (20344.74) ●	3.00 (2019600.00) ○	1.00 (226.00) ●	3.00 (67828155.12) ○
		MGM-2	3.90 (396097.17) ○	3.00 (52573.82) ○	4.00 (2914990.41) ○	4.00 (817.00) ○	4.00 (90626441.02) ○
90%	30%	COOPT	1.00 (198970.22) ●	4.00 (1768646.10) ○	2.00 (1793702.00) ●	2.50 (232.00)	2.00 (52030639.92) ●
		DSA	2.00 (200493.11) ●	2.00 (44303.00) ●	1.00 (1770000.00) ●	2.50 (232.00)	1.00 (43831090.97) ●
		MGM	3.20 (223577.47) ○	1.00 (36551.18) ●	3.00 (3651600.00) ○	1.00 (226.00) ●	3.00 (122583653.52) ○
		MGM-2	3.80 (224952.53) ○	3.00 (97205.10) ○	4.00 (5267938.10) ○	4.00 (817.00) ○	4.00 (163036967.97) ○
	50%	COOPT	1.00 (365359.58) ●	4.00 (1768646.10) ○	2.00 (1793702.00) ●	2.50 (232.00)	2.00 (52010506.00) ●
		DSA	2.00 (367481.02) ●	2.00 (44293.56) ●	1.00 (1770000.00) ●	2.50 (232.00)	1.00 (43823748.39) ●
		MGM	3.10 (395236.95) ○	1.00 (36576.80) ●	3.00 (3651600.00) ○	1.00 (226.00) ●	3.00 (122583238.24) ○
		MGM-2	3.90 (396800.37) ○	3.00 (96660.73) ○	4.00 (5267988.90) ○	4.00 (817.00) ○	4.00 (163047872.83) ○
	90%	COOPT	1.00 (724361.12) ●	4.00 (1769860.86) ○	2.00 (1793702.00) ●	2.50 (232.00)	2.00 (51975937.52) ●
		DSA	2.00 (726349.70) ●	2.00 (44228.25) ●	1.00 (1770000.00) ●	2.50 (232.00)	1.00 (43818981.62) ●
		MGM	3.30 (753810.28) ○	1.00 (36564.48) ●	3.00 (3651600.00) ○	1.00 (226.00) ●	3.00 (122582894.56) ○
		MGM-2	3.70 (755325.27) ○	3.00 (97590.77) ○	4.00 (5267969.90) ○	4.00 (817.00) ○	4.00 (163056494.00) ○

Tabela 5.8: Resultados dos experimentos envolvendo problemas com 200 variáveis em redes aleatórias (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (50796.06) ●	4.00 (585857.90) ○	2.00 (604562.00) ●	2.50 (232.00)	2.00 (17620912.08) ●
		DSA	2.00 (51947.52) ●	2.00 (19217.75) ●	1.00 (597000.00) ●	2.50 (232.00)	1.00 (14614073.03) ●
		MGM	3.00 (53651.58) ○	1.00 (15961.27) ●	3.00 (1217880.00) ○	1.00 (226.00) ●	3.00 (40922186.72) ○
		MGM-2	4.00 (54244.38) ○	3.00 (35729.31) ○	4.00 (1758354.15) ○	4.00 (817.00) ○	4.00 (54813127.30) ○
	50%	COOPT	1.00 (102595.74) ●	4.00 (584935.10) ○	2.00 (604562.00) ●	2.50 (232.00)	2.00 (17612834.12) ●
		DSA	2.00 (104082.34) ●	2.00 (18905.79) ●	1.00 (597000.00) ●	2.50 (232.00)	1.00 (14614618.63) ●
		MGM	3.20 (106545.15) ○	1.00 (15708.73) ●	3.00 (1217880.00) ○	1.00 (226.00) ●	3.00 (40922922.16) ○
		MGM-2	3.80 (106876.87) ○	3.00 (35191.60) ○	4.00 (1758368.04) ○	4.00 (817.00) ○	4.00 (54812134.97) ○
	90%	COOPT	1.00 (221857.46) ●	4.00 (585562.90) ○	2.00 (604562.00) ●	2.50 (232.00)	2.00 (17601574.86) ●
		DSA	2.00 (223221.76) ●	2.00 (19245.01) ●	1.00 (597000.00) ●	2.50 (232.00)	1.00 (14614305.75) ●
		MGM	3.10 (225476.30) ○	1.00 (15967.62) ●	3.00 (1217880.00) ○	1.00 (226.00) ●	3.00 (40922969.80) ○
		MGM-2	3.90 (226133.20) ○	3.00 (35507.84) ○	4.00 (1758439.09) ○	4.00 (817.00) ○	4.00 (54829858.41) ○
50%	30%	COOPT	1.00 (97577.16) ●	4.00 (981343.40) ○	2.00 (995602.00) ●	2.50 (232.00)	2.00 (29093995.22) ●
		DSA	2.00 (99204.59) ●	2.00 (29288.97) ●	1.00 (994000.00) ●	2.50 (232.00)	1.00 (24359447.40) ●
		MGM	3.20 (106850.80) ○	1.00 (24288.50) ●	3.00 (2029800.00) ○	1.00 (226.00) ●	3.00 (68169949.08) ○
		MGM-2	3.80 (107285.71) ○	3.00 (56101.98) ●	4.00 (2929553.24) ○	4.00 (817.00) ○	4.00 (91026854.85) ○
	50%	COOPT	1.00 (188554.60) ●	4.00 (979359.10) ○	2.00 (995602.00) ●	2.50 (232.00)	2.00 (29078924.92) ●
		DSA	2.00 (189940.35) ●	2.00 (29682.68) ●	1.00 (994000.00) ●	2.50 (232.00)	1.00 (24358019.78) ●
		MGM	3.20 (198936.76) ○	1.00 (24610.69) ●	3.00 (2029800.00) ○	1.00 (226.00) ●	3.00 (68169458.08) ○
		MGM-2	3.80 (199493.64) ○	3.00 (56500.60) ○	4.00 (2929667.65) ○	4.00 (817.00) ○	4.00 (91066148.30) ○
	90%	COOPT	1.00 (387836.14) ●	4.00 (980192.80) ○	2.00 (995602.00) ●	2.50 (232.00)	2.00 (29069380.22) ●
		DSA	2.00 (389795.41) ●	2.00 (29398.01) ●	1.00 (994000.00) ●	2.50 (232.00)	1.00 (24354459.43) ●
		MGM	3.40 (398829.80) ○	1.00 (24429.73) ●	3.00 (2029800.00) ○	1.00 (226.00) ●	3.00 (68169296.52) ○
		MGM-2	3.60 (398947.97) ○	3.00 (56498.35) ○	4.00 (2929645.46) ○	4.00 (817.00) ○	4.00 (91057818.37) ○
90%	30%	COOPT	1.10 (198915.60) ●	4.00 (1768369.80) ○	2.00 (1794682.00) ●	2.50 (232.00)	2.00 (52044935.88) ●
		DSA	1.90 (200721.82) ●	2.00 (46710.21) ●	1.00 (1771000.00) ●	2.50 (232.00)	1.00 (43848773.05) ●
		MGM	3.10 (223147.53) ○	1.00 (38691.27) ●	3.00 (3653640.00) ○	1.00 (226.00) ●	3.00 (122651575.76) ○
		MGM-2	3.90 (224582.20) ○	3.00 (98280.50) ○	4.00 (5270898.50) ○	4.00 (817.00) ○	4.00 (163125240.03) ○
	50%	COOPT	1.00 (365707.44) ●	4.00 (1768368.10) ○	2.00 (1794682.00) ●	2.50 (232.00)	2.00 (52037588.94) ●
		DSA	2.00 (367786.07) ●	2.00 (46774.71) ●	1.00 (1771000.00) ●	2.50 (232.00)	1.00 (43851294.94) ●
		MGM	3.10 (395129.59) ○	1.00 (38736.45) ●	3.00 (3653640.00) ○	1.00 (226.00) ●	3.00 (122651867.48) ○
		MGM-2	3.90 (396777.67) ○	3.00 (98804.37) ○	4.00 (5270889.97) ○	4.00 (817.00) ○	4.00 (163125443.43) ○
	90%	COOPT	1.00 (725407.36) ●	4.00 (1769182.90) ○	2.00 (1794682.00) ●	2.50 (232.00)	2.00 (52003064.90) ●
		DSA	2.00 (727383.33) ●	2.00 (46816.14) ●	1.00 (1771000.00) ●	2.50 (232.00)	1.00 (43855836.64) ●
		MGM	3.10 (754539.21) ○	1.00 (38801.16) ●	3.00 (3653640.00) ○	1.00 (226.00) ●	3.00 (122651969.48) ○
		MGM-2	3.90 (756197.27) ○	3.00 (97467.50) ○	4.00 (5270956.37) ○	4.00 (817.00) ○	4.00 (163151674.60) ○

Tabela 5.9: Resultados dos experimentos envolvendo problemas com 200 variáveis em redes de mundos pequenos (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (49122.26) ●	4.00 (572306.30) ○	2.00 (587902.00) ●	2.50 (232.00)	2.00 (17132515.14) ●
		DSA	2.00 (50124.67) ●	2.00 (17090.94) ●	1.00 (580000.00) ●	2.50 (232.00)	1.00 (14199448.81) ●
		MGM	3.20 (51977.79) ○	1.00 (14204.24) ●	3.00 (1183200.00) ○	1.00 (226.00) ●	3.00 (39757278.04) ○
		MGM-2	3.80 (52276.26) ○	3.00 (32346.02) ○	4.00 (1708232.66) ○	4.00 (817.00) ○	4.00 (53233143.95) ○
	50%	COOPT	1.00 (99235.54) ●	4.00 (570705.40) ○	2.00 (587902.00) ●	2.50 (232.00)	2.00 (17122128.88) ●
		DSA	2.00 (100610.75) ●	2.00 (17202.14) ●	1.00 (580000.00) ●	2.50 (232.00)	1.00 (14201319.60) ●
		MGM	3.00 (102584.58) ○	1.00 (14284.97) ●	3.00 (1183200.00) ○	1.00 (226.00) ●	3.00 (39758646.72) ○
		MGM-2	4.00 (103003.09) ○	3.00 (32630.47) ○	4.00 (1708344.59) ○	4.00 (817.00) ○	4.00 (53267256.36) ○
	90%	COOPT	1.00 (214896.34) ●	4.00 (567480.20) ○	2.00 (587902.00) ●	2.50 (232.00)	2.00 (17114077.68) ●
		DSA	2.00 (216372.24) ●	2.00 (17444.56) ●	1.00 (580000.00) ●	2.50 (232.00)	1.00 (14196161.25) ●
		MGM	3.40 (218485.11) ○	1.00 (14480.45) ●	3.00 (1183200.00) ○	1.00 (226.00) ●	3.00 (39758694.96) ○
		MGM-2	3.60 (218752.41) ○	3.00 (32763.54) ○	4.00 (1708414.79) ○	4.00 (817.00) ○	4.00 (53285550.29) ○
50%	30%	COOPT	1.00 (95900.58) ●	4.00 (965873.10) ○	2.00 (989902.00) ●	2.50 (232.00)	2.00 (28665197.08) ●
		DSA	2.00 (97219.08) ●	2.00 (27648.38) ●	1.00 (970000.00) ●	2.50 (232.00)	1.00 (23991781.80) ●
		MGM	3.30 (104780.67) ○	1.00 (22941.06) ●	3.00 (1999200.00) ○	1.00 (226.00) ●	3.00 (67144119.68) ○
		MGM-2	3.70 (105266.31) ○	3.00 (53963.90) ○	4.00 (2885417.22) ○	4.00 (817.00) ○	4.00 (89668554.05) ○
	50%	COOPT	1.00 (185074.70) ●	4.00 (963000.80) ○	2.00 (989902.00) ●	2.50 (232.00)	2.00 (28654232.04) ●
		DSA	2.00 (186822.05) ●	2.00 (27827.07) ●	1.00 (970000.00) ●	2.50 (232.00)	1.00 (23990301.86) ●
		MGM	3.10 (195453.08) ○	1.00 (23075.46) ●	3.00 (1999200.00) ○	1.00 (226.00) ●	3.00 (67143352.16) ○
		MGM-2	3.90 (196417.19) ○	3.00 (54308.60) ○	4.00 (2885477.87) ○	4.00 (817.00) ○	4.00 (89680763.06) ○
	90%	COOPT	1.00 (380667.04) ●	4.00 (966879.50) ○	2.00 (989902.00) ●	2.50 (232.00)	2.00 (28627056.08) ●
		DSA	2.00 (382675.59) ●	2.00 (27984.68) ●	1.00 (970000.00) ●	2.50 (232.00)	1.00 (23992092.98) ●
		MGM	3.30 (391723.04) ○	1.00 (23204.01) ●	3.00 (1999200.00) ○	1.00 (226.00) ●	3.00 (67143243.20) ○
		MGM-2	3.70 (392135.59) ○	3.00 (54224.20) ○	4.00 (2885516.00) ○	4.00 (817.00) ○	4.00 (89689271.39) ○
90%	30%	COOPT	1.00 (197374.86) ●	4.00 (1759093.70) ○	2.00 (1783902.00) ●	2.50 (232.00)	2.00 (51731639.32) ●
		DSA	2.00 (199221.10) ●	2.00 (47079.35) ●	1.00 (1760000.00) ●	2.50 (232.00)	1.00 (43579136.09) ●
		MGM	3.40 (222220.17) ○	1.00 (39007.67) ●	3.00 (3631200.00) ○	1.00 (226.00) ●	3.00 (121900046.27) ○
		MGM-2	3.60 (222697.10) ○	3.00 (98043.50) ○	4.00 (5238483.90) ○	4.00 (817.00) ○	4.00 (162109072.97) ○
	50%	COOPT	1.00 (362740.10) ●	4.00 (1757070.60) ○	2.00 (1783902.00) ●	2.50 (232.00)	2.00 (51713448.30) ●
		DSA	2.00 (364523.68) ●	2.00 (47190.70) ●	1.00 (1760000.00) ●	2.50 (232.00)	1.00 (43572713.85) ●
		MGM	3.20 (391625.97) ○	1.00 (39109.83) ●	3.00 (3631200.00) ○	1.00 (226.00) ●	3.00 (121901117.13) ○
		MGM-2	3.80 (393159.67) ○	3.00 (98052.80) ○	4.00 (5238513.10) ○	4.00 (817.00) ○	4.00 (162127046.20) ○
	90%	COOPT	1.00 (719550.18) ●	4.00 (1756284.10) ○	2.00 (1783902.00) ●	2.50 (232.00)	2.00 (51680180.66) ●
		DSA	2.00 (721602.17) ●	2.00 (47096.92) ●	1.00 (1760000.00) ●	2.50 (232.00)	1.00 (43573235.73) ●
		MGM	3.30 (749137.33) ○	1.00 (39007.43) ●	3.00 (3631200.00) ○	1.00 (226.00) ●	3.00 (121900939.93) ○
		MGM-2	3.70 (750767.37) ○	3.00 (98305.07) ○	4.00 (5238553.93) ○	4.00 (817.00) ○	4.00 (162133891.07) ○

Tabela 5.10: Resultados dos experimentos envolvendo problemas com 200 variáveis em redes livres de escala (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.10 (45872.16) ●	4.00 (407579.30) ○	2.00 (545272.00) ●	2.50 (232.00)	2.00 (15857335.34) ●
		DSA	1.90 (47272.00) ●	2.00 (44618.80) ●	1.00 (536500.00) ●	2.50 (232.00)	1.00 (13101793.68) ●
		MGM	3.50 (48534.99) ○	1.00 (37007.03) ●	3.00 (1094460.00) ○	1.00 (226.00) ●	3.00 (36763427.08) ○
		MGM-2	3.50 (48582.75) ○	3.00 (98533.59) ○	4.00 (1580131.57) ○	4.00 (817.00) ○	4.00 (49283455.03) ○
	50%	COOPT	1.00 (93186.36) ●	4.00 (404163.60) ○	2.00 (545272.00) ●	2.50 (232.00)	2.00 (15853724.10) ●
		DSA	2.00 (95336.83) ●	2.00 (44436.00) ●	1.00 (536500.00) ●	2.50 (232.00)	1.00 (13101426.31) ●
		MGM	3.40 (96552.77) ○	1.00 (36856.46) ●	3.00 (1094460.00) ○	1.00 (226.00) ●	3.00 (36765218.84) ○
		MGM-2	3.60 (96692.37) ○	3.00 (98683.79) ○	4.00 (1580313.43) ○	4.00 (817.00) ○	4.00 (49337268.48) ○
	90%	COOPT	1.00 (200631.02) ●	4.00 (403323.96) ○	2.00 (545272.00) ●	2.50 (232.00)	2.00 (15838983.92) ●
		DSA	2.00 (202296.52) ●	2.00 (44295.89) ●	1.00 (536500.00) ●	2.50 (232.00)	1.00 (13100592.56) ●
		MGM	3.10 (203443.97) ○	1.00 (36742.78) ●	3.00 (1094460.00) ○	1.00 (226.00) ●	3.00 (36765915.52) ○
		MGM-2	3.90 (203894.98) ○	3.00 (97431.33) ○	4.00 (1580445.44) ○	4.00 (817.00) ○	4.00 (49361041.65) ○
50%	30%	COOPT	1.00 (83446.74) ●	4.00 (666377.50) ○	2.00 (859852.00) ●	2.50 (232.00)	2.00 (25115263.72) ●
		DSA	2.00 (84770.68) ●	2.00 (47502.59) ●	1.00 (857500.00) ●	2.50 (232.00)	1.00 (20953601.22) ●
		MGM	3.30 (87953.07) ○	1.00 (39359.18) ●	3.00 (1749300.00) ○	1.00 (226.00) ●	3.00 (58757547.84) ○
		MGM-2	3.70 (88165.21) ○	3.00 (104155.74) ○	4.00 (2525026.96) ○	4.00 (817.00) ○	4.00 (78594837.23) ○
	50%	COOPT	1.00 (161047.28) ●	4.00 (666323.40) ○	2.00 (859852.00) ●	2.50 (232.00)	2.00 (25106028.42) ●
		DSA	2.00 (163080.23) ●	2.00 (47322.68) ●	1.00 (857500.00) ●	2.50 (232.00)	1.00 (20954288.78) ●
		MGM	3.20 (166864.13) ○	1.00 (39218.24) ●	3.00 (1749300.00) ○	1.00 (226.00) ●	3.00 (58756545.56) ○
		MGM-2	3.80 (167725.87) ○	3.00 (104011.66) ○	4.00 (2525172.59) ○	4.00 (817.00) ○	4.00 (78641859.32) ○
	90%	COOPT	1.00 (332242.14) ●	4.00 (673362.90) ○	2.00 (859852.00) ●	2.50 (232.00)	2.00 (25090104.58) ●
		DSA	2.00 (334388.18) ●	2.00 (47520.03) ●	1.00 (857500.00) ●	2.50 (232.00)	1.00 (20956076.16) ●
		MGM	3.20 (338420.56) ○	1.00 (39382.99) ●	3.00 (1749300.00) ○	1.00 (226.00) ●	3.00 (58756022.52) ○
		MGM-2	3.80 (338970.84) ○	3.00 (104243.77) ○	4.00 (2525240.01) ○	4.00 (817.00) ○	4.00 (78660080.73) ○
90%	30%	COOPT	1.00 (146883.96) ●	4.00 (1134967.66) ○	2.00 (1379412.00) ●	2.50 (232.00)	2.00 (40177914.18) ●
		DSA	2.00 (148562.62) ●	2.00 (48896.41) ●	1.00 (1371500.00) ●	2.50 (232.00)	1.00 (33740080.11) ●
		MGM	3.10 (162402.07) ○	1.00 (40598.53) ●	3.00 (2814180.00) ○	1.00 (226.00) ●	3.00 (94477498.47) ○
		MGM-2	3.90 (164159.27) ○	3.00 (105717.67) ○	4.00 (4060847.60) ○	4.00 (817.00) ○	4.00 (125985260.27) ○
	50%	COOPT	1.00 (274795.62) ●	4.00 (1126065.30) ○	2.00 (1379412.00) ●	2.50 (232.00)	2.00 (40158817.28) ●
		DSA	2.00 (276854.37) ●	2.00 (48944.65) ●	1.00 (1371500.00) ●	2.50 (232.00)	1.00 (33735124.40) ●
		MGM	3.30 (293135.40) ○	1.00 (40622.10) ●	3.00 (2814180.00) ○	1.00 (226.00) ●	3.00 (94479715.87) ○
		MGM-2	3.70 (294495.27) ○	3.00 (104969.40) ○	4.00 (4060801.70) ○	4.00 (817.00) ○	4.00 (125952436.20) ○
	90%	COOPT	1.00 (550407.50) ●	4.00 (1140366.90) ○	2.00 (1379412.00) ●	2.50 (232.00)	2.00 (40122744.86) ●
		DSA	2.00 (552649.34) ●	2.00 (48941.98) ●	1.00 (1371500.00) ●	2.50 (232.00)	1.00 (33740982.23) ●
		MGM	3.60 (570579.37) ○	1.00 (40625.63) ●	3.00 (2814180.00) ○	1.00 (226.00) ●	3.00 (94477304.73) ○
		MGM-2	3.40 (571453.00) ○	3.00 (106570.13) ○	4.00 (4060895.07) ○	4.00 (817.00) ○	4.00 (125993106.20) ○

Já nos experimentos envolvendo os maiores e mais complexos problemas, isto é, com 200 variáveis, o COOPT obteve as melhores soluções em todos os cenários. Contudo, os resultados demonstraram uma leve variação na topologia de rede livres de escala. Em sua essência, o modelo de Kuramoto descreve a sincronização em redes envolvendo osciladores iguais ou muito similares em uma topologia globalmente conectada. A sensibilidade do modelo de Kuramoto em redes livres de escala é algo conhecido e bastante estudado (ARENAS et al., 2008). Este comportamento ocorre devido à grande variação na distribuição de grau da rede, sobretudo em função do volume de elementos com baixo grau. Esta característica surge como um desafio para a definição de uma força de acoplamento capaz de alcançar um estado estável durante o processo de sincronização.

#### 5.4.2 Ciclos de Mensagens

O COOPT, assim como os outros algoritmos de busca local avaliados, requer uma quantidade linear de mensagens de acordo com o limite  $n$  de *rounds* de comunicação predefinido. Por sua vez, o DPOP resultou no menor volume de mensagens compartilhadas dos experimentos devido à técnica de programação dinâmica utilizada para calcular os custos agregados em cada subárvore do problema. No entanto, o tamanho da maior mensagem enviada pelos agentes torna-se exponencial no DPOP.

Por sua vez, no DSA cada agente envia mensagens para seus vizinhos a fim de informá-los sobre novas atribuições em cada *round* de comunicação. Adicionalmente, no COOPT os agentes também notificam os agentes filhos quando deve ser iniciado um novo estágio de sincronização e informam o agente pai sobre o custo da solução parcial iniciada na subárvore do agente atual. Por sua vez, o MGM e o MGM-2 requerem mais mensagens que o DSA e o COOPT devido às mensagens adicionais utilizadas para evitar ótimos locais. Por fim, a técnica de amostragem introduzida nos agentes do DUCT demandou o maior volume de mensagens enviadas entre os experimentos.

#### 5.4.3 NCCC

Como já comentado na Seção 4.2.1, o COOPT requer uma ordenação prévia dos agentes para assegurar visões locais consistentes durante os *rounds* de comunicação e para oferecer uma busca local *anytime*. Este mecanismo evita computações locais irrelevantes devido às visões locais desatualizadas sobre o problema e, conseqüentemente, acelera o processo de convergência da qualidade das soluções encontradas durante o processo de sincronização. Além disto, a técnica de busca local *anytime* contribui para a estabilização das soluções encontradas em cada *round* de comunicação.

Não obstante, o número de verificações não concorrentes cresce mais rapidamente no COOPT ao comparar com os demais os algoritmos avaliados. Isto ocorre devido ao comportamento síncrono dos agentes, quando o contador  $NCCC^a$  é atualizado quando um agente recebe uma dada mensagem. Por outro lado, a maioria dos algoritmos avaliados recebem mensagens e realizam as computações locais de maneira assíncrona e, com isto, o contador  $NCCC^a$  cresce sobretudo a partir de verificações de restrições realizadas pelos agentes. O COOPT apresentou uma leve melhora no NCCC em redes livres de escala, justificado pela existência de *hubs* na rede, isto é, elementos altamente conectados. Nestes cenários, os *rounds* de comunicação são dominados pelos *hubs* da rede.

#### 5.4.4 Tamanho das Mensagens e Informações Compartilhadas

Todos os algoritmos de busca local avaliados apresentam um tamanho constante para a maior mensagem enviada, quando cada agente controla uma única variável. No COOPT os agentes enviam informações em mensagens do tipo *assign*, as quais contêm os novos valores de suas variáveis e fases depois de executar o estágio de sincronização. Também são enviadas mensagens do tipo *cost* e *synchronize*, as quais contêm a informação do custo acumulado da solução parcial da subárvore iniciada no agente atual, além da identificação do *round* com a melhor solução global até o momento, respectivamente. De maneira similar, os agentes do DSA e do MGM compartilham suas visões locais em cada ciclo de mensagem. Por outro lado, no MGM-2 os agentes compartilham as utilidades das atribuições de grupos de agentes, produzindo mensagens maiores.

Para os algoritmos DPOP e no DUCT, o tamanho da maior mensagem pode ser exponencial em função do número de variáveis e densidade do grafo de restrições. Deste modo, devido ao baixo volume de mensagens compartilhadas entre os agentes e ao tamanho constante para a maior mensagem, o COOPT também resultou em um custo satisfatório de comunicação, em relação ao total de informações compartilhadas entre os agentes. Este comportamento foi observado em todos cenários avaliados. Por sua vez, o DSA não apresentou diferença estatisticamente significativa sobre esta métrica em relação ao COOPT. Por fim, o algoritmo MGM-2 apresentou o pior desempenho nesta métrica em relação aos demais algoritmos de busca local, devido ao tamanho das mensagens e da quantidade de mensagens compartilhadas entre os agentes.

## 5.5 Análise de Convergência

De fato, uma das características mais relevantes do modelo de Kuramoto é sua capacidade de convergência, a qual apresenta uma tendência natural para lidar com redes complexas, incluindo diferentes padrões de conexão entre os elementos. Esta capacidade de convergência, aliada ao processo iterativo de estabilização coletiva, proporcionam uma grande escalabilidade e aplicabilidade deste modelo de sincronização física. Não obstante, é fundamental que estratégias para resolução de DCOP baseadas em sincronização física capturem esta notável característica do modelo de Kuramoto. Neste sentido, além do comparativo de desempenho do COOPT, também é importante avaliar sua capacidade e velocidade de convergência envolvendo problemas complexos.

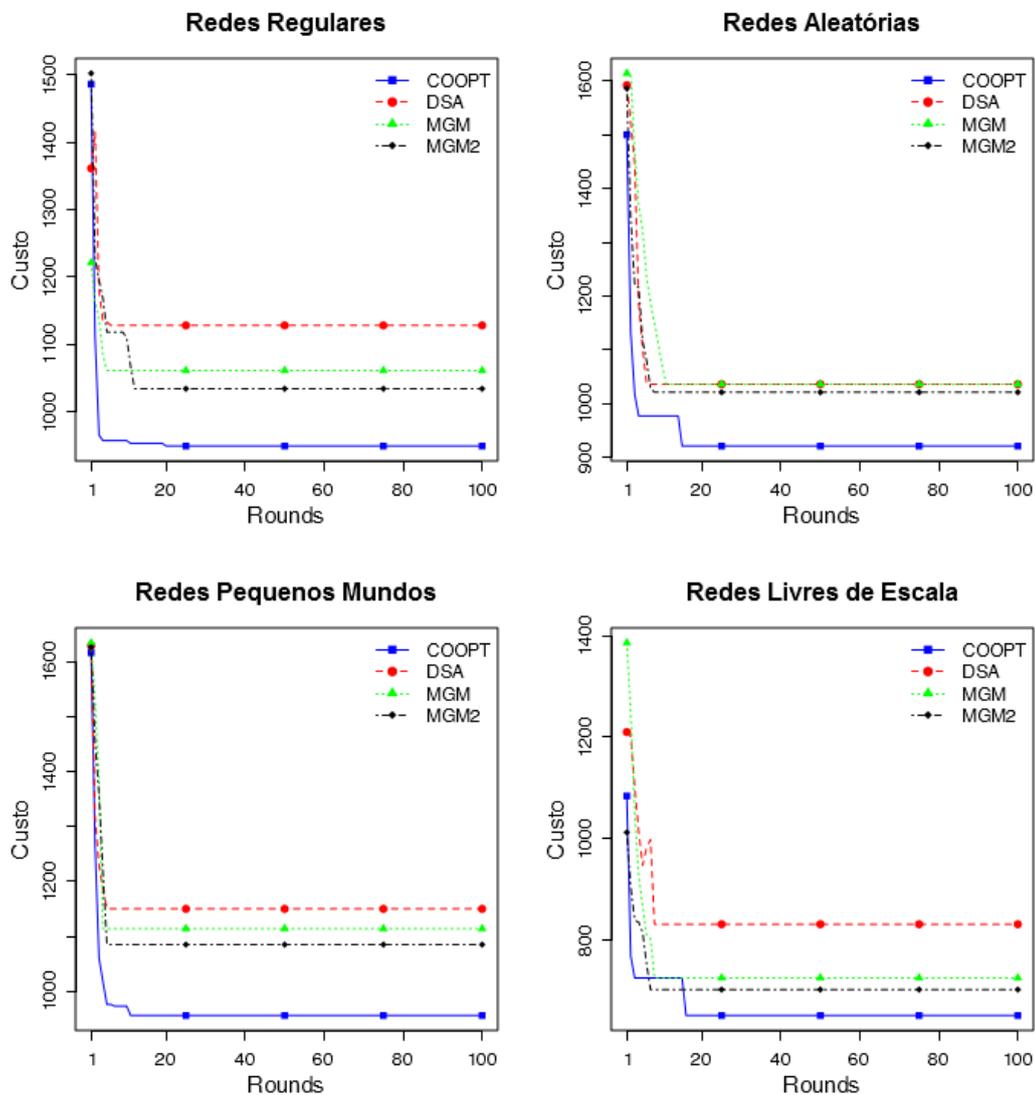


Figura 5.2: Comparativo da convergência dos algoritmos de busca local nos experimentos envolvendo 10 variáveis e 90% de densidade e *tightness*.

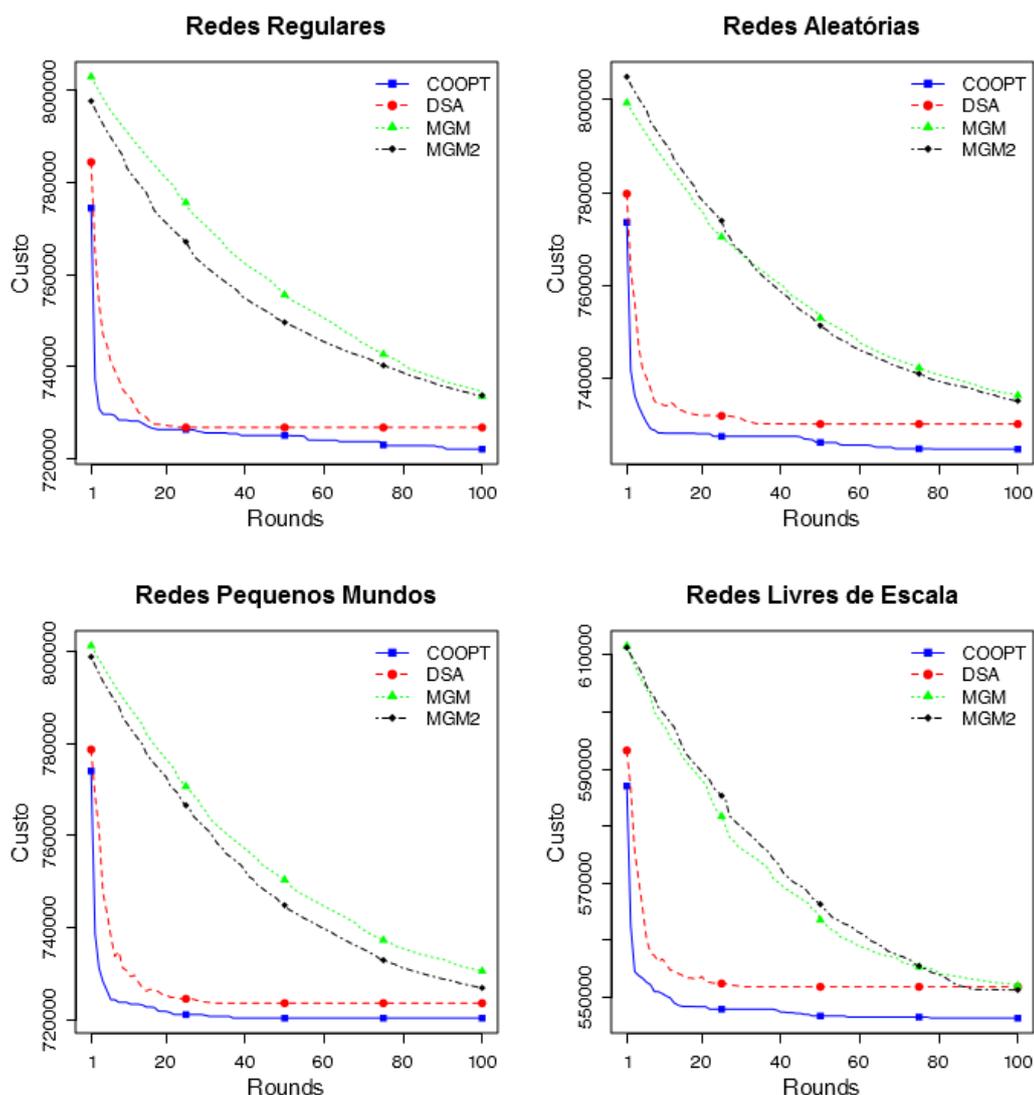


Figura 5.3: Comparativo da convergência dos algoritmos de busca local nos experimentos envolvendo 200 variáveis e 90% de densidade e *tightness*.

As Figuras 5.2 e 5.3 apresentam uma análise de convergência dos custos das soluções ao longo do tempo, cuja unidade de tempo é representada por um *round* de comunicação. Nesta análise foi comparada a capacidade de convergência dos algoritmos de busca local avaliados, considerando um limite de 100 *rounds*. O comparativo foi realizado com base nos problemas de maior complexidade descritos na Seção 5.1, envolvendo 10 e 200 variáveis e 90% de densidade e de *tightness*. Os resultados desta análise de convergência foram separados em topologias de redes regulares, aleatórias, mundos pequenos e livres de escala, possibilitando também a análise entre diferentes padrões de conexão da rede sobre a capacidade de convergência dos algoritmos de busca local avaliados.

A partir dos resultados retratados pela Figuras 5.2 e 5.3, é possível observar que o COOPT apresenta uma convergência da qualidade das soluções mais eficiente em relação

aos demais algoritmos de busca local. Por sua vez, o MGM e o MGM-2 apresentaram uma significativa degradação na velocidade de convergência em problemas envolvendo 200 variáveis. Este comportamento é decorrente do mecanismo de otimização monotônico implementado por estes algoritmos. Já o DSA possui um mecanismo de otimização não monotônico, portanto, além de não garantir a melhoria das soluções em *rounds* futuros, por vezes a busca atinge ótimos locais que, conseqüentemente, interrompe o processo de convergência. No entanto, ainda assim o DSA apresentou uma velocidade de convergência superior ao MGM e MGM-2.

Além disto, a capacidade de convergência do COOPT manteve um comportamento estável entre as diferentes topologias de rede avaliadas, embora os melhores resultados tenham sido observados em redes regulares e livres de escala. Esta análise também revelou que a convergência para soluções de alta qualidade ocorreu sobretudo em menos de 20 *rounds* de comunicação. Após este limite, os ganhos na qualidade das soluções foram praticamente irrelevantes frente ao esforço computacional demandado pelos *rounds* adicionais. Esta é uma das principais vantagens de algoritmos *anytime* como o COOPT.

## 5.6 Análise do *Tightness*

Durante a análise experimental, também foi mensurado o impacto de aumentar o percentual de *tightness* nos problemas sobre o desempenho dos algoritmos. Esta análise se justifica pois, em geral, problemas mais restritivos requerem mais *rounds* para alcançar soluções de alta qualidade. Problemas envolvendo 30% de *tightness* possuem um certo grau de dificuldade, porém, ainda assim são satisfatórios. Por outro lado, problemas com 50% ou mais de *tightness* apresentam um alto grau de dificuldade, visto que a grande maioria das restrições são insatisfatórias (GRINSHPOUN et al., 2013).

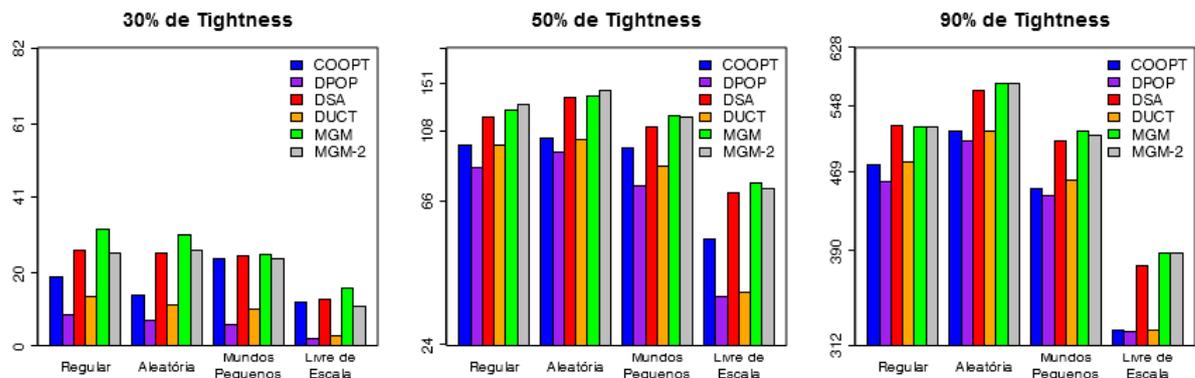


Figura 5.4: Análise sobre impacto do *tightness* sobre o custo das soluções nos experimentos envolvendo 10 variáveis.

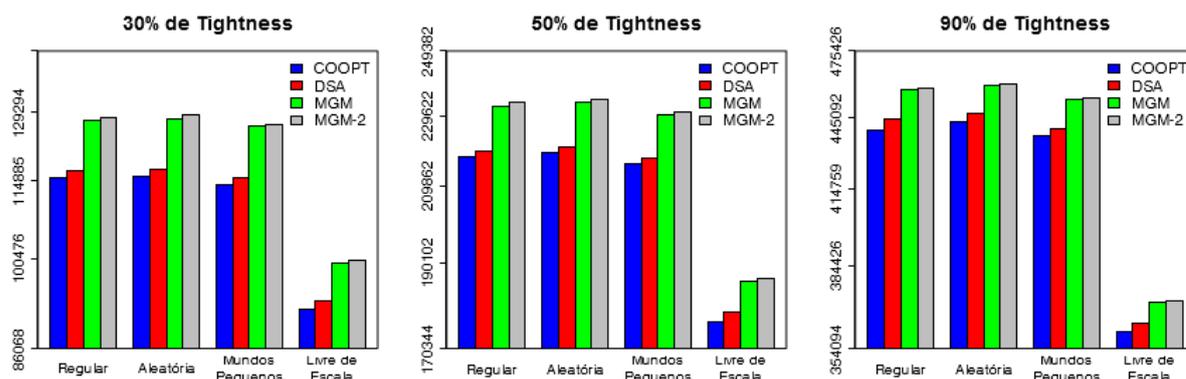


Figura 5.5: Análise sobre impacto do *tightness* sobre o custo das soluções nos experimentos envolvendo 200 variáveis.

As Figuras 5.4 e 5.5 apresentam uma comparação do custo médio das soluções obtidas em cada algoritmo avaliado, considerando problemas com 10 variáveis e 200 restrições em diferentes topologias de rede. De forma geral, os resultados indicaram que o aumento do *tightness* não apresentou um impacto significativo sobre o desempenho dos algoritmos avaliados, no que tange à qualidade das soluções encontradas. No entanto, foi possível observar o COOPT teve seu desempenho ligeiramente degradado em problemas envolvendo poucas variáveis e um baixo *tightness*.

Além disto, nos experimentos realizados sobre os menores problemas menores (envolvendo 10 variáveis), o COOPT produziu soluções de alta qualidade em problemas complexos (com 90% de *tightness*) em relação aos demais algoritmos avaliados. Já em cenários em larga escala (envolvendo 200 variáveis), todos os algoritmos de busca local avaliados apresentaram um crescimento linear do custo das soluções encontradas conforme aumentado do percentual de *tightness* dos problemas. No entanto, o COOPT produziu os melhores resultados em comparação aos demais algoritmos de busca local, mantendo um comportamento consistente entre as diferentes topologias de rede abordadas.

## 5.7 Considerações Finais

Os experimentos demonstraram que o COOPT é capaz de lidar com problemas em larga escala mais eficientemente que os demais algoritmos avaliados. O COOPT alcançou os melhores resultados em relação ao número de mensagens e *throughput* de rede. Neste sentido, o COOPT apresentou melhorias significativas em comunicação perante os demais algoritmos avaliados, demonstrando um modelo de comunicação eficiente. Este modelo de comunicação e a computação local dos agentes proporciona uma rápida convergência para soluções de alta qualidade, em especial em ambientes complexos e em larga escala.

Além disto, o COOPT também apresentou os melhores resultados em termos da qualidade das soluções encontradas, observados principalmente durante o crescimento da quantidade de variáveis, restrições e *tightness* dos problemas envolvidos nos experimentos. Em outras palavras, conforme o tamanho e a densidade dos grafos aumentava durante os experimentos, foram observadas soluções com qualidade cada vez mais alta em comparação aos demais algoritmos avaliados. Esta observação demonstrou a tendência natural do COOPT para lidar com problemas densos, justificada pelo reforço da atualização das fases pelas variáveis vizinhas durante o processo de sincronização física.

Os experimentos também revelaram que as topologias de redes avaliadas não apresentaram um efeito significativo sobre a eficiência do COOPT. Contudo, o número de variáveis, o grau médio quanto a distribuição de grau são propriedades de rede que podem influenciar na capacidade de convergência do COOPT. Em problemas esparsos, isto é, com uma baixa densidade, o desempenho do COOPT pode ser ligeiramente afetado. Portanto, o tamanho dos problemas, a densidade e a distribuição de grau da rede são fatores que podem contribuir para um desempenho satisfatório do COOPT.

Finalmente, as mesmas conclusões são mantidas sobre os resultados das demais métricas. A quantidade limite de *rounds* também é um critério fundamental para o desempenho do COOPT, bem como para os demais algoritmos de busca local avaliados. A partir da análise de convergência realizada, recomenda-se a utilização de 20 *rounds* para equilibrar a relação entre soluções de alta qualidade e esforço computacional demandado pelo processo de busca. Estes fatores favorecem a utilização do COOPT como modelo de coordenação descentralizada em sistemas multiagente complexos e em larga escala.

## Capítulo 6

### Conclusão

Algoritmos para o Problema de Otimização Distribuída de Restrição (DCOP) dispõem de um conjunto de técnicas eficientes para coordenação em sistemas multiagente. Recentemente este campo de pesquisa tem recebido muita atenção devido à capacidade de modelar naturalmente um grande número de problemas reais por meio de um conjunto de variáveis e de restrições. As motivações para a utilização deste formalismo para coordenação descentralizada em sistemas multiagente refere-se ao fato dos métodos para resolução de DCOP serem essencialmente distribuídos e robustos.

No entanto, devido ao fato do DCOP pertencer à classe de problemas NP-completo, alguns cenários reais não são favoráveis aos métodos atuais para resolução de DCOP, em especial ao se tratar de ambientes complexos e em larga escala, envolvendo centenas de variáveis ou restrições. Em geral, as limitações dos métodos atuais para resolução de DCOP estão relacionadas à falta de escalabilidade e ao esforço computacional requerido pelos algoritmos em função do tamanho do espaço de estados do problema.

Nos últimos anos, o foco principal das pesquisas sobre DCOP esteve centrado no desenvolvimento de novos métodos de resolução. Com isto, diversos algoritmos para DCOP foram introduzidos na literatura até então, contudo, ainda permaneceram em aberto questões sobre a escalabilidade dos algoritmos. Um dos aspectos que contribui para esta limitação está relacionada à baixa diversidade teórica dos algoritmos atuais para DCOP, resultando em estratégias de resolução com desempenho similares.

Por sua vez, modelos de sincronização física em redes de osciladores acoplados têm sido intensamente estudados nas últimas décadas em virtude das notáveis habilidades de convergência, escalabilidade e aplicabilidade em diversos cenários de problemas reais. Neste sentido, a investigação sobre modelos sincronização constituiu um dos principais objetivos deste trabalho, visando a concepção de uma nova abordagem para resolução de DCOP baseada em modelos de sincronização física em redes de osciladores acoplados,

capaz de lidar com ambientes complexos e em larga escala.

Mais especificamente, este trabalho introduziu um novo algoritmo para DCOP, denominado *Coupled Oscillators OPTimization* (COOPT), cuja estratégia de busca é inspirada em princípios da sincronização física descritos pelo modelo de Kuramoto. A estratégia de coordenação descentralizada do COOPT consiste em convergir gradualmente o custo local de uma solução parcial do problema para um custo mínimo global entre grupos de agentes, levando em consideração o impacto das ações locais sobre os vizinhos por meio de restrições entre as variáveis.

Desta maneira, no COOPT cada agente será influenciado pelos vizinhos a escolher um valor para sua variável que aproxime o custo da solução ao mínimo global entre grupos de agentes, ao invés de escolher um valor que reduza exclusivamente o custo local da variável. Esta influência sobre as ações vizinhas é representada por uma aproximação de erro presente no modelo de Kuramoto, a qual refere-se à fase de um oscilador. Por sua vez, esta aproximação é realizada levando em consideração a similaridade do erro entre os agentes vizinhos.

Para mensurar o desempenho do COOPT, foram realizados diversos experimentos a partir de instâncias de DCOP geradas aleatoriamente, mas que reproduzem as principais características presentes em ambientes complexos reais e em larga escala. Desta maneira, a avaliação realizada levou em consideração diferentes níveis de complexidade nos problemas ao introduzir nos experimentos diferentes topologias de redes complexas e níveis de restrição das custos de custo. Os experimentos também mensuraram a escalabilidade ao considerar diferentes tamanhos do espaço de estados e densidade nos problemas.

Portanto, as métricas utilizadas na avaliação experimental do COOPT foram: i) a quantidade de ciclos de mensagens; ii) NCCC; iii) tamanho da maior mensagem compartilhada entre agentes; iv) *throughput* de rede; e v) custo global das soluções encontradas. Além disto, os resultados obtidos pelo COOPT foram comparados com outros algoritmos para DCOP que compõem o estado da arte deste campo de pesquisa, sendo eles o DPOP, DUCT, DSA, MGM e o MGM-2. O DPOP é um algoritmo completo, ao passo que o DUCT, DSA, MGM e MGM-2 são incompletos. Além disto, DSA, MGM e MGM-2 são algoritmos de busca local.

Os resultados dos experimentos demonstraram que o COOPT é capaz de lidar com problemas complexos e em larga escala de maneira mais eficiente que os demais algoritmos avaliados. O COOPT apresentou, em média, os melhores resultados em relação a qualidade das soluções encontradas, número de mensagens enviadas e *throughput* de rede. Estes resultados são justificados sobretudo pela complexidade algorítmica linear do COOPT. Outra característica observada no COOPT refere-se a rápida de convergência

para soluções de alta qualidade, principalmente em problemas complexos e em larga escala. Este comportamento demonstra a tendência natural do COOPT para tratar especialmente problemas densos.

Os experimentos também revelaram que as topologias de redes não apresentaram um efeito significativo sobre a eficiência do COOPT. Contudo, os resultados demonstraram que a distribuição de grau impacta na capacidade de convergência do COOPT. Também foi notado que a eficiência do COOPT é ligeiramente afetada em problemas pequenos e esparsos. Nestas condições, o COOPT apresentou um comportamento menos estável, embora ainda assim tenha alcançado os melhores resultados em todos os cenários, em termos de qualidade das soluções. Portanto, tanto o tamanho, densidade e distribuição de grau da rede são fatores relevantes para um desempenho satisfatório do COOPT.

Neste sentido, foi observado experimentalmente que é possível capturar a essência dos modelos de sincronização física e utilizá-la como técnica para resolução de DCOP. Adicionalmente, as características mais relevantes presentes em modelos de sincronização física, tais como escalabilidade e a habilidade de convergência, também foram observadas na avaliação experimental do COOPT, o que reforça o potencial desta fundamentação teórica para pesquisas futuras sobre DCOP.

## 6.1 Trabalhos Futuros

Os resultados observados pelo método para resolução de DCOP proposto por este trabalho sugerem que, de fato, novas investigações sobre modelos de sincronização física podem ser realizadas para impulsionar a evolução deste campo de pesquisa. No entanto, embora os resultados obtidos sejam encorajadores, algumas perspectivas futuras merecem investigações mais específicas e detalhadas. Recomenda-se, por exemplo, uma análise e compreensão mais profunda do impacto da força global de acoplamento  $K$  sobre a capacidade de convergência da qualidade das soluções, com o propósito de resolver ou suavizar algumas das limitações atuais do COOPT.

Deste modo, estudos visando a compreensão mais aprofundada das implicações do acoplamento global  $K$  sobre o processo de sincronização pode ser um viés de pesquisa para aprimorar a eficiência do algoritmo em redes esparsas, ou quando os elementos da rede apresentam uma grande variação na distribuição de grau. Assim, sugere-se novas investigações sobre o acoplamento global  $K$  levando em consideração diferentes aspectos ou propriedades de uma rede complexa. Outra recomendação é um estudo analítico sobre a sensibilidade da aproximação de erro local dos agentes e a eficiência do modelo de comunicação em função de  $K$ , bem como uma forma para determiná-lo adequadamente.

Além disto, os estudos deste trabalho enfatizaram formulações de DCOP prevendo ambientes estáticos, bem como um comportamento e conhecimento determinístico dos agentes. Assim, sugere-se que as vantagens observadas no COOPT, sobretudo a habilidade de convergência em ambientes complexos e em larga escala, também possa contribuir com os desafios presentes em formulações alternativas de DCOP (discutidas na Seção 2.4). Em outras palavras, a velocidade de convergência do método proposto, por exemplo, pode ser útil para cenários dinâmicos (D-DCOP) ou probabilísticos (P-DCOP), os quais requerem uma capacidade de adaptação em resposta às mudanças no ambiente percebidas pelos agentes ao longo do tempo.

Por fim, alguns estudos complementares, como a análise do impacto da estrutura de priorização dos agentes sobre o processo de busca do COOPT, bem como a concepção de uma condição de término alternativa capaz de oferecer garantia das soluções encontradas, também poderiam contribuir para ampliar a aplicabilidade do algoritmo em uma maior variedade de problemas complexos reais. Estas são, em suma, algumas sugestões para trabalhos futuros nesta área de pesquisa, em especial, relacionados a escalabilidade dos métodos para resolução de DCOP.

# Capítulo 7

## Apêndice

Tabela 7.1: Resultados dos experimentos envolvendo problemas com 50 variáveis em redes regulares (CD: 1.93)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (1100.96) ●	4.00 (34771.09) ○	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1107622.30) ●
		DSA	2.30 (1252.72) ●	2.00 (3583.56) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854227.25) ●
		DUCT	5.00 (2304.87) ○	5.00 (5370773.33) ○	5.00 (5203673.31) ○	4.00 (473.87) ○	5.00 (885485135.28) ○
		MGM	3.50 (1373.57) ○	1.00 (2936.12) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2398062.83)
		MGM-2	3.20 (1339.40) ○	3.00 (7203.99)	4.00 (103290.61) ○	5.00 (817.00) ○	4.00 (3254914.43) ○
	50%	COOPT	1.00 (3466.50) ●	4.00 (35255.66) ●	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1107774.08) ●
		DSA	2.20 (3740.71) ●	2.00 (3586.24) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854562.80) ●
		DUCT	5.00 (5321.67) ○	5.00 (5402817.84) ○	5.00 (5198504.70) ○	4.00 (472.10) ○	5.00 (881269081.68) ○
		MGM	3.10 (3884.60) ○	1.00 (2922.36) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2398214.56)
		MGM-2	3.70 (3900.69) ○	3.00 (7167.84)	4.00 (103339.74) ○	5.00 (816.58) ○	4.00 (3264768.44) ○
	90%	COOPT	1.00 (10243.37) ●	4.00 (35019.40) ○	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1107094.24) ●
		DSA	2.10 (10529.14) ●	2.00 (3598.82) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854509.04) ●
DUCT		5.00 (12216.85) ○	5.00 (5366032.56) ○	5.00 (5201413.06) ○	4.00 (470.20) ○	5.00 (879008749.99) ○	
MGM		3.70 (10690.97) ○	1.00 (2912.95) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2398217.92)	
MGM-2		3.20 (10647.43) ○	3.00 (7161.06)	4.00 (103424.11) ○	5.00 (816.58) ○	4.00 (3279706.89) ○	
50%	30%	COOPT	1.00 (3124.81) ●	4.00 (60499.20) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1829309.60) ●
		DSA	2.20 (3409.05) ●	2.00 (6042.14) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1465434.72) ●
		DUCT	5.00 (5208.63) ○	5.00 (9856056.96) ○	5.00 (5325961.26) ○	4.00 (518.10) ○	5.00 (964874491.35) ○
		MGM	3.70 (3578.18) ○	1.00 (4964.27) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111257.60)
		MGM-2	3.10 (3534.10) ○	3.00 (12318.13)	4.00 (176839.41) ○	5.00 (816.58) ○	4.00 (5535107.06) ○
	50%	COOPT	1.00 (7861.63) ●	4.00 (60744.00) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1827909.92) ●
		DSA	2.10 (8171.89) ●	2.00 (6035.71) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1465022.16) ●
		DUCT	5.00 (10585.38) ○	5.00 (9815136.96) ○	5.00 (5309779.84) ○	4.00 (514.40) ○	5.00 (957724872.08) ○
		MGM	3.40 (8396.25) ○	1.00 (4966.20) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111238.40)
		MGM-2	3.50 (8392.42) ○	3.00 (12321.75)	4.00 (176877.91) ○	5.00 (817.00) ○	4.00 (5545035.38) ○
	90%	COOPT	1.00 (19716.77) ●	4.00 (60988.80) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1826934.56) ●
		DSA	2.30 (20075.12) ●	2.00 (6024.81) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1465044.24) ●
DUCT		5.00 (22527.12) ○	5.00 (9866014.08) ○	5.00 (5317784.06) ○	4.00 (518.30) ○	5.00 (960204276.84) ○	
MGM		3.40 (20225.47) ○	1.00 (4958.24) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111359.36)	
MGM-2		3.30 (20229.63) ○	3.00 (12282.24)	4.00 (176919.93) ○	5.00 (817.00) ○	4.00 (5552823.04) ○	
90%	30%	COOPT	1.10 (8174.74) ●	4.00 (111387.60) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3270294.88) ●
		DSA	1.90 (8480.98) ●	2.00 (10853.85) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2686809.84) ●
		DUCT	5.00 (11487.76) ○	5.00 (18692131.94) ○	5.00 (5380032.92) ○	4.00 (553.00) ○	5.00 (994881012.33) ○
		MGM	3.70 (8873.76) ○	1.00 (8985.15) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537168.32)
		MGM-2	3.30 (8795.65) ○	3.00 (22588.98)	4.00 (324001.92) ○	5.00 (817.00) ○	4.00 (10098392.18) ○
	50%	COOPT	1.00 (17568.71) ●	4.00 (111167.20) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3268211.04) ●
		DSA	2.20 (17983.97) ●	2.00 (10855.91) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2686014.76) ●
		DUCT	5.00 (21343.71) ○	5.00 (18612092.42) ○	5.00 (5369990.04) ○	4.00 (553.00) ○	5.00 (992738093.99) ○
		MGM	3.20 (18287.30) ○	1.00 (8986.11) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537562.56)
		MGM-2	3.60 (18305.01) ○	3.00 (22633.28)	4.00 (324015.45) ○	5.00 (817.00) ○	4.00 (10101721.28) ○
	90%	COOPT	1.00 (39568.50) ●	4.00 (110946.80) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3266669.28) ●
		DSA	2.20 (40016.34) ●	2.00 (10859.37) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2686178.88) ●
DUCT		5.00 (43648.74) ○	5.00 (18604916.80) ○	5.00 (5376774.90) ○	4.00 (550.10) ○	5.00 (993779010.46) ○	
MGM		3.30 (40241.87) ○	1.00 (8983.01) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537713.92)	
MGM-2		3.50 (40268.60) ○	3.00 (22542.30)	4.00 (324047.61) ○	5.00 (817.00) ○	4.00 (10109540.82) ○	

Tabela 7.2: Resultados dos experimentos envolvendo problemas com 50 variáveis em redes aleatórias (CD: 1.93)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (1270.45) ●	4.00 (35731.00) ○	2.00 (40768.00) ●	3.00 (232.00)	2.00 (1156721.20) ●
		DSA	2.20 (1458.34) ●	2.00 (5293.19) ●	1.00 (36700.00) ●	2.00 (230.00) ●	1.00 (895787.60) ●
		DUCT	5.00 (2496.22) ○	5.00 (4245370.84) ○	5.00 (5061324.28) ○	4.00 (433.30) ○	5.00 (799459044.72) ○
		MGM	3.60 (1577.51) ○	1.00 (4392.45) ●	3.00 (74868.00)	1.00 (224.00) ●	3.00 (2514459.84)
	50%	MGM-2	3.20 (1576.13) ○	3.00 (9481.25)	4.00 (108301.66) ○	5.00 (816.58) ○	4.00 (3413290.51) ○
		COOPT	1.00 (3748.40) ●	4.00 (36087.90) ○	2.00 (40768.00) ●	3.00 (232.00)	2.00 (1156357.74) ●
		DSA	2.10 (4023.36) ●	2.00 (5466.50) ●	1.00 (36700.00) ●	2.00 (230.00) ●	1.00 (895876.56) ●
		DUCT	5.00 (5413.32) ○	5.00 (4233317.44) ○	5.00 (5038483.68) ○	4.00 (430.40) ○	5.00 (796993842.03) ○
	90%	MGM	3.60 (4237.27) ○	1.00 (4533.64) ●	3.00 (74868.00)	1.00 (224.00) ●	3.00 (2514555.16)
		MGM-2	3.30 (4167.40) ○	3.00 (9643.17)	4.00 (108351.99) ○	5.00 (816.58) ○	4.00 (3420151.64) ○
		COOPT	1.00 (11022.77) ●	4.00 (35920.30) ○	2.00 (40768.00) ●	3.00 (232.00)	2.00 (1156236.20) ●
		DSA	2.20 (11268.93) ●	2.00 (5322.72) ●	1.00 (36700.00) ●	2.00 (230.00) ●	1.00 (895818.50) ●
50%	30%	DUCT	5.00 (12827.51) ○	5.00 (4036172.88) ○	5.00 (5013204.84) ○	4.00 (430.30) ○	5.00 (780606654.93) ○
		MGM	3.40 (11439.63) ○	1.00 (4413.02) ●	3.00 (74868.00)	1.00 (224.00) ●	3.00 (2514651.68)
		MGM-2	3.40 (11473.93) ○	3.00 (9459.00)	4.00 (108426.41) ○	5.00 (817.00) ○	4.00 (3430950.91) ○
		COOPT	1.00 (3304.70) ●	4.00 (60727.70) ○	2.00 (64778.00) ●	3.00 (232.00)	2.00 (1863801.30) ●
	50%	DSA	2.30 (3587.27) ●	2.00 (8135.48) ●	1.00 (61200.00) ●	2.00 (230.00) ●	1.00 (1494309.44) ●
		DUCT	5.00 (5348.10) ○	5.00 (7861496.70) ○	5.00 (5274273.24) ○	4.00 (495.50) ○	5.00 (931088201.31) ○
		MGM	3.60 (3736.91) ○	1.00 (6759.62) ●	3.00 (124848.00)	1.00 (224.00) ●	3.00 (4193472.68)
		MGM-2	3.10 (3697.68) ○	3.00 (14616.15)	4.00 (180376.62) ○	5.00 (817.00) ○	4.00 (5647368.96) ○
	90%	COOPT	1.00 (8098.95) ●	4.00 (61005.60) ○	2.00 (64778.00) ●	3.00 (232.00)	2.00 (1861897.62) ●
		DSA	2.00 (8423.67) ●	2.00 (8013.96) ●	1.00 (61200.00) ●	2.00 (230.00) ●	1.00 (1493864.89) ●
		DUCT	5.00 (10640.02) ○	5.00 (7637711.96) ○	5.00 (5245378.78) ○	4.00 (485.10) ○	5.00 (919278524.05) ○
		MGM	3.60 (8621.22) ○	1.00 (6655.06) ●	3.00 (124848.00)	1.00 (224.00) ●	3.00 (4193532.68)
90%	30%	MGM-2	3.40 (8592.12) ○	3.00 (14454.08)	4.00 (180411.08) ○	5.00 (816.58) ○	4.00 (5653186.54) ○
		COOPT	1.00 (19933.32) ●	4.00 (60105.50) ○	2.00 (64778.00) ●	3.00 (232.00)	2.00 (1861414.20) ●
		DSA	2.00 (20327.36) ●	2.00 (8107.75) ●	1.00 (61200.00) ●	2.00 (230.00) ●	1.00 (1494240.63) ●
		DUCT	5.00 (22736.49) ○	5.00 (7816467.76) ○	5.00 (5242088.22) ○	4.00 (490.70) ○	5.00 (917778570.79) ○
	50%	MGM	3.70 (20566.24) ○	1.00 (6727.34) ●	3.00 (124848.00)	1.00 (224.00) ●	3.00 (4193800.00)
		MGM-2	3.30 (20565.55) ○	3.00 (14715.49)	4.00 (180466.91) ○	5.00 (817.00) ○	4.00 (5666807.63) ○
		COOPT	1.10 (8310.07) ●	4.00 (111164.60) ○	2.00 (112798.00) ●	3.00 (232.00)	2.00 (3273953.48) ●
		DSA	2.10 (8659.84) ●	2.00 (11874.92) ●	1.00 (110200.00) ●	2.00 (230.00) ●	1.00 (2691228.90) ●
	90%	DUCT	5.00 (11525.50) ○	5.00 (16972956.36) ○	5.00 (5377132.20) ○	4.00 (541.20) ○	5.00 (992071680.77) ○
		MGM	3.10 (8906.65) ○	1.00 (9840.02) ●	3.00 (224808.00)	1.00 (224.00) ●	3.00 (7550670.08)
		MGM-2	3.70 (8933.08) ○	3.00 (23149.66)	4.00 (324577.19) ○	5.00 (816.58) ○	4.00 (10113036.77) ○
		COOPT	1.00 (17596.57) ●	4.00 (110539.40) ○	2.00 (112798.00) ●	3.00 (232.00)	2.00 (3273213.80) ●
50%	DSA	2.00 (18043.20) ●	2.00 (11848.25) ●	1.00 (110200.00) ●	2.00 (230.00) ●	1.00 (2691787.01) ●	
	DUCT	5.00 (21462.88) ○	5.00 (16708341.22) ○	5.00 (5361716.62) ○	4.00 (540.30) ○	5.00 (988342673.18) ○	
	MGM	3.60 (18462.69) ○	1.00 (9819.97) ●	3.00 (224808.00)	1.00 (224.00) ●	3.00 (7550842.32)	
	MGM-2	3.40 (18396.38) ○	3.00 (23107.42)	4.00 (324603.16) ○	5.00 (817.00) ○	4.00 (10121256.96) ○	
90%	COOPT	1.00 (39722.11) ●	4.00 (111350.70) ○	2.00 (112798.00) ●	3.00 (232.00)	2.00 (3272372.96) ●	
	DSA	2.00 (40134.22) ●	2.00 (11872.77) ●	1.00 (110200.00) ●	2.00 (230.00) ●	1.00 (2691632.56) ●	
	DUCT	5.00 (43528.90) ○	5.00 (16589383.90) ○	5.00 (5369958.54) ○	4.00 (541.50) ○	5.00 (990451350.18) ○	
	MGM	3.40 (40461.17) ○	1.00 (9842.12) ●	3.00 (224808.00)	1.00 (224.00) ●	3.00 (7551107.44)	
MGM-2	3.60 (40474.40) ○	3.00 (23065.83)	4.00 (324633.67) ○	5.00 (816.58) ○	4.00 (10128364.76) ○		

Tabela 7.3: Resultados dos experimentos envolvendo problemas com 50 variáveis em redes de mundos pequenos (CD: 1.93)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (1105.83) ●	4.00 (34594.52) ○	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1107401.08) ●
		DSA	2.00 (1288.76) ●	2.00 (4583.09) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854198.63) ●
		DUCT	5.00 (2324.21) ○	5.00 (4222612.88) ○	5.00 (5064989.20) ○	4.00 (430.60) ○	5.00 (788308900.05) ○
		MGM	3.50 (1390.63) ○	1.00 (3813.65) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2397963.04)
	50%	MGM-2	3.50 (1398.92) ○	3.00 (8271.62)	4.00 (103285.10) ○	5.00 (817.00) ○	4.00 (3253736.87) ○
		COOPT	1.00 (3343.92) ●	4.00 (35056.00) ○	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1107702.68) ●
		DSA	2.10 (3571.16) ●	2.00 (4831.10) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854167.36) ●
		DUCT	5.00 (5156.42) ○	5.00 (4261251.52) ○	5.00 (5118037.16) ○	4.00 (453.90) ○	5.00 (822381948.04) ○
	90%	MGM	3.60 (3767.99) ○	1.00 (4007.10) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2398225.12)
		MGM-2	3.30 (3728.57) ○	3.00 (8581.14)	4.00 (103343.77) ○	5.00 (816.58) ○	4.00 (3265116.28) ○
		COOPT	1.00 (10205.53) ●	4.00 (34218.60) ○	2.00 (39102.00) ●	3.00 (232.00)	2.00 (1106836.42) ●
		DSA	2.10 (10513.52) ●	2.00 (4665.64) ●	1.00 (35000.00) ●	2.00 (230.00) ●	1.00 (854384.47) ●
50%	30%	DUCT	5.00 (12156.22) ○	5.00 (4112414.92) ○	5.00 (5043331.76) ○	4.00 (432.40) ○	5.00 (783285585.61) ○
		MGM	3.20 (10661.16) ○	1.00 (3871.42) ●	3.00 (71400.00)	1.00 (224.00) ●	3.00 (2398201.48)
		MGM-2	3.70 (10682.22) ○	3.00 (8387.07)	4.00 (103416.41) ○	5.00 (814.90) ○	4.00 (3276203.05) ○
		COOPT	1.00 (3223.12) ●	4.00 (59915.30) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1829185.26) ●
	50%	DSA	2.10 (3466.57) ●	2.00 (7495.11) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1465288.55) ●
		DUCT	5.00 (5233.15) ○	5.00 (8169247.46) ○	5.00 (5275620.40) ○	4.00 (491.70) ○	5.00 (925880561.19) ○
		MGM	3.40 (3635.96) ○	1.00 (6257.23) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111137.72)
		MGM-2	3.50 (3662.48) ○	3.00 (13606.99)	4.00 (176831.29) ○	5.00 (817.00) ○	4.00 (5533046.97) ○
	90%	COOPT	1.00 (7790.05) ●	4.00 (59711.80) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1827885.66) ●
		DSA	2.00 (8142.89) ●	2.00 (7353.63) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1465016.03) ●
		DUCT	5.00 (10408.92) ○	5.00 (7922313.40) ○	5.00 (5265039.56) ○	4.00 (496.00) ○	5.00 (931195444.70) ○
		MGM	3.60 (8352.09) ○	1.00 (6130.46) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111386.44)
90%	30%	MGM-2	3.40 (8296.56) ○	3.00 (13389.81)	4.00 (176888.83) ○	5.00 (817.00) ○	4.00 (5547723.51) ○
		COOPT	1.00 (19710.15) ●	4.00 (60126.60) ○	2.00 (63602.00) ●	3.00 (232.00)	2.00 (1827135.94) ●
		DSA	2.00 (20020.98) ●	2.00 (7494.77) ●	1.00 (60000.00) ●	2.00 (230.00) ●	1.00 (1464869.91) ●
		DUCT	5.00 (22391.77) ○	5.00 (7987033.28) ○	5.00 (5278342.16) ○	4.00 (496.60) ○	5.00 (932850648.44) ○
	50%	MGM	3.60 (20278.21) ○	1.00 (6239.32) ●	3.00 (122400.00)	1.00 (224.00) ●	3.00 (4111378.04)
		MGM-2	3.40 (20207.77) ○	3.00 (13541.98)	4.00 (176926.18) ○	5.00 (816.58) ○	4.00 (5553900.62) ○
		COOPT	1.10 (8253.78) ●	4.00 (111159.60) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3270048.58) ●
		DSA	2.05 (8593.39) ●	2.00 (11877.94) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2686704.53) ●
	90%	DUCT	5.00 (11472.33) ○	5.00 (16182235.14) ○	5.00 (5374967.46) ○	4.00 (544.40) ○	5.00 (991622856.28) ○
		MGM	3.35 (8809.77) ○	1.00 (9840.95) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537462.84)
		MGM-2	3.50 (8775.33) ○	3.00 (23093.06)	4.00 (324004.59) ○	5.00 (817.00) ○	4.00 (10099042.59) ○
		COOPT	1.00 (17354.54) ●	4.00 (110784.40) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3268467.72) ●
90%	DSA	2.20 (17754.63) ●	2.00 (11778.49) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2687019.59) ●	
	DUCT	5.00 (21360.14) ○	5.00 (16032338.68) ○	5.00 (5363170.42) ○	4.00 (541.20) ○	5.00 (988682813.30) ○	
	MGM	3.30 (18088.01) ○	1.00 (9761.49) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537565.20)	
	MGM-2	3.50 (18166.32) ○	3.00 (23003.48)	4.00 (324017.36) ○	5.00 (817.00) ○	4.00 (10103142.35) ○	
90%	COOPT	1.00 (39716.69) ●	4.00 (110594.90) ○	2.00 (112602.00) ●	3.00 (232.00)	2.00 (3267412.14) ●	
	DSA	2.00 (40108.48) ●	2.00 (11826.37) ●	1.00 (110000.00) ●	2.00 (230.00) ●	1.00 (2686396.66) ●	
	DUCT	5.00 (43682.28) ○	5.00 (16383768.72) ○	5.00 (5372631.06) ○	4.00 (544.10) ○	5.00 (991113287.07) ○	
	MGM	3.50 (40577.63) ○	1.00 (9798.46) ●	3.00 (224400.00)	1.00 (224.00) ●	3.00 (7537386.52)	
MGM-2	3.50 (40568.93) ○	3.00 (23076.45)	4.00 (324031.95) ○	5.00 (816.16) ○	4.00 (10105503.22) ○		

Tabela 7.4: Resultados dos experimentos envolvendo problemas com 50 variáveis em redes livres de escala (CD: 1.93)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.20 (1043.74) ●	4.00 (24958.40) ○	2.00 (36358.00) ●	3.00 (232.00)	2.00 (1026039.26) ●
		DSA	2.00 (1212.34) ●	2.00 (10095.99) ●	1.00 (32200.00) ●	2.00 (230.00) ●	1.00 (784623.05) ●
		DUCT	4.80 (1398.86) ○	5.00 (1377242.46) ○	5.00 (2960539.66) ○	4.00 (418.00) ○	5.00 (291447190.57) ○
		MGM	3.30 (1309.14) ○	1.00 (8373.10) ●	3.00 (65688.00) ●	1.00 (224.00) ●	3.00 (2205466.20) ●
		MGM-2	3.70 (1298.18) ○	3.00 (18594.12) ●	4.00 (95124.53) ○	5.00 (817.00) ○	4.00 (3002332.62) ○
	50%	COOPT	1.00 (3328.93) ●	4.00 (24989.90) ○	2.00 (36358.00) ●	3.00 (232.00)	2.00 (1026127.80) ●
		DSA	2.00 (3565.08) ●	2.00 (10076.70) ●	1.00 (32200.00) ●	2.00 (230.00) ●	1.00 (784745.90) ●
		DUCT	4.80 (3910.70) ○	5.00 (1373885.52) ○	5.00 (2992742.28) ○	4.00 (418.00) ○	5.00 (296588189.20) ○
		MGM	3.90 (3759.14) ○	1.00 (8354.94) ●	3.00 (65688.00) ●	1.00 (224.00) ●	3.00 (2205450.72) ●
		MGM-2	3.30 (3689.14) ○	3.00 (18467.54) ●	4.00 (95251.96) ○	5.00 (816.58) ○	4.00 (3019845.04) ○
	90%	COOPT	1.00 (9342.27) ●	4.00 (24663.20) ○	2.00 (36358.00) ●	3.00 (232.00)	2.00 (1025538.34) ●
		DSA	2.20 (9752.96) ●	2.00 (9881.50) ●	1.00 (32200.00) ●	2.00 (230.00) ●	1.00 (784876.36) ●
DUCT		4.80 (10146.15) ○	5.00 (1342667.62) ○	5.00 (2937541.16) ○	4.00 (418.00) ○	5.00 (283378839.13) ○	
MGM		3.10 (9874.68) ○	1.00 (8187.61) ●	3.00 (65688.00) ●	1.00 (224.00) ●	3.00 (2205789.64) ●	
MGM-2		3.90 (9906.29) ○	3.00 (17549.30) ●	4.00 (95350.52) ○	5.00 (816.58) ○	4.00 (3035247.33) ○	
50%	30%	COOPT	1.00 (2685.28) ●	4.00 (42013.80) ○	2.00 (55958.00) ●	3.00 (232.00)	2.00 (1602660.44) ●
		DSA	2.10 (2961.58) ●	2.00 (11403.79) ●	1.00 (52200.00) ●	2.00 (230.00) ●	1.00 (1272681.20) ●
		DUCT	5.00 (3464.36) ○	5.00 (2723648.27) ○	5.00 (3602495.12) ○	4.00 (418.00) ○	5.00 (419554642.51) ○
		MGM	3.50 (3127.89) ○	1.00 (9452.70) ●	3.00 (106488.00) ●	1.00 (224.00) ●	3.00 (3575630.80) ●
		MGM-2	3.40 (3102.75) ○	3.00 (21300.48) ●	4.00 (153934.52) ○	5.00 (816.58) ○	4.00 (4827253.79) ○
	50%	COOPT	1.00 (6621.78) ●	4.00 (41361.70) ○	2.00 (55958.00) ●	3.00 (232.00)	2.00 (1602189.40) ●
		DSA	2.10 (6965.74) ●	2.00 (11279.43) ●	1.00 (52200.00) ●	2.00 (230.00) ●	1.00 (1272928.26) ●
		DUCT	5.00 (7763.71) ○	5.00 (2807139.76) ○	5.00 (3655413.18) ○	4.00 (418.00) ○	5.00 (433084931.24) ○
		MGM	3.60 (7188.69) ○	1.00 (9355.79) ●	3.00 (106488.00) ●	1.00 (224.00) ●	3.00 (3575787.12) ●
		MGM-2	3.30 (7162.99) ○	3.00 (21043.86) ●	4.00 (153996.67) ○	5.00 (816.58) ○	4.00 (4839607.75) ○
	90%	COOPT	1.00 (16938.58) ●	4.00 (41610.80) ○	2.00 (55958.00) ●	3.00 (232.00)	2.00 (1601267.74) ●
		DSA	2.20 (17288.00) ●	2.00 (11422.45) ●	1.00 (52200.00) ●	2.00 (230.00) ●	1.00 (1273045.41) ●
DUCT		5.00 (18176.31) ○	5.00 (2915261.04) ○	5.00 (3681502.42) ○	4.00 (418.00) ○	5.00 (442321762.52) ○	
MGM		3.40 (17479.38) ○	1.00 (9479.96) ●	3.00 (106488.00) ●	1.00 (224.00) ●	3.00 (3576089.40) ●	
MGM-2		3.40 (17511.71) ○	3.00 (20960.61) ●	4.00 (154073.45) ○	5.00 (815.66) ○	4.00 (4852524.32) ○	
90%	30%	COOPT	1.00 (5566.91) ●	4.00 (70348.10) ○	2.00 (87808.00) ●	3.00 (232.00)	2.00 (2539558.52) ●
		DSA	2.20 (5851.96) ●	2.00 (12064.83) ●	1.00 (84700.00) ●	2.00 (230.00) ●	1.00 (2067548.74) ●
		DUCT	5.00 (7246.39) ○	5.00 (6418955.02) ○	5.00 (4454328.30) ○	4.00 (418.00) ○	5.00 (646097817.93) ○
		MGM	3.70 (6083.53) ○	1.00 (9999.36) ●	3.00 (172788.00) ●	1.00 (224.00) ●	3.00 (5803220.08) ●
		MGM-2	3.10 (6049.37) ○	3.00 (23121.76) ●	4.00 (249545.09) ○	5.00 (817.00) ○	4.00 (7790987.55) ○
	50%	COOPT	1.00 (12769.11) ●	4.00 (72438.60) ○	2.00 (87808.00) ●	3.00 (232.00)	2.00 (2539171.52) ●
		DSA	2.00 (13177.80) ●	2.00 (12063.16) ●	1.00 (84700.00) ●	2.00 (230.00) ●	1.00 (2068082.66) ●
		DUCT	5.00 (14980.12) ○	5.00 (6710644.82) ○	5.00 (4567240.06) ○	4.00 (418.00) ○	5.00 (684264929.31) ○
		MGM	3.40 (13422.66) ○	1.00 (9999.30) ●	3.00 (172788.00) ●	1.00 (224.00) ●	3.00 (5803281.68) ●
		MGM-2	3.60 (13477.11) ○	3.00 (23171.27) ●	4.00 (249586.65) ○	5.00 (817.00) ○	4.00 (7802255.34) ○
	90%	COOPT	1.00 (29712.06) ●	4.00 (70823.00) ○	2.00 (87808.00) ●	3.00 (232.00)	2.00 (2538244.50) ●
		DSA	2.40 (30127.11) ●	2.00 (12063.19) ●	1.00 (84700.00) ●	2.00 (230.00) ●	1.00 (2067641.96) ●
DUCT		5.00 (31980.41) ○	5.00 (6430847.14) ○	5.00 (4513455.36) ○	4.00 (418.00) ○	5.00 (672110664.66) ○	
MGM		3.20 (30307.45) ○	1.00 (10000.35) ●	3.00 (172788.00) ●	1.00 (224.00) ●	3.00 (5803467.96) ●	
MGM-2		3.40 (30305.46) ○	3.00 (23147.69) ●	4.00 (249617.53) ○	5.00 (816.58) ○	4.00 (7806273.81) ○	

Tabela 7.5: Resultados dos experimentos envolvendo problemas com 100 variáveis em redes regulares (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (8556.33) ●	4.00 (144430.01) ○	2.00 (151802.00) ●	3.00 (232.00) ○	2.00 (4380498.57) ●
		DSA	2.00 (9084.38) ●	2.00 (7377.74) ●	1.00 (145000.00) ●	2.00 (230.00) ●	1.00 (3542589.03) ●
		MGM	3.60 (9543.70) ○	1.00 (6015.19) ●	3.00 (295800.00) ○	1.00 (224.00) ●	3.00 (9935920.24) ○
		MGM-2	3.40 (9487.88) ○	3.00 (15038.61) ○	4.00 (427316.23) ○	4.00 (817.00) ○	4.00 (13374542.20) ○
	50%	COOPT	1.00 (20426.43) ●	4.00 (143274.10) ○	2.00 (151802.00) ●	3.00 (232.00) ○	2.00 (4378788.96) ●
		DSA	2.00 (20986.48) ●	2.00 (7371.45) ●	1.00 (145000.00) ●	2.00 (230.00) ●	1.00 (3542747.30) ●
		MGM	3.60 (21600.13) ○	1.00 (6013.81) ●	3.00 (295800.00) ○	1.00 (224.00) ●	3.00 (9936193.44) ○
		MGM-2	3.40 (21524.10) ○	3.00 (14993.61) ○	4.00 (427356.82) ○	4.00 (817.00) ○	4.00 (13381348.98) ○
	90%	COOPT	1.00 (49369.54) ●	4.00 (144596.20) ○	2.00 (151802.00) ●	3.00 (232.00) ○	2.00 (4376308.88) ●
		DSA	2.10 (49968.23) ●	2.00 (7358.24) ●	1.00 (145000.00) ●	2.00 (230.00) ●	1.00 (3543303.81) ●
		MGM	3.60 (50496.35) ○	1.00 (6008.92) ●	3.00 (295800.00) ○	1.00 (224.00) ●	3.00 (9936870.88) ○
		MGM-2	3.30 (50455.26) ○	3.00 (15011.15) ○	4.00 (427424.58) ○	4.00 (817.00) ○	4.00 (13393533.10) ○
50%	30%	COOPT	1.00 (19171.98) ●	4.00 (243529.20) ○	2.00 (249802.00) ●	3.00 (232.00) ○	2.00 (7263083.36) ●
		DSA	2.00 (19798.81) ●	2.00 (12296.43) ●	1.00 (245000.00) ●	2.00 (230.00) ●	1.00 (5987468.27) ●
		MGM	3.30 (20358.69) ○	1.00 (10075.01) ●	3.00 (499800.00) ○	1.00 (224.00) ●	3.00 (16789677.52) ○
		MGM-2	3.70 (20417.51) ○	3.00 (25456.85) ○	4.00 (721600.57) ○	4.00 (817.00) ○	4.00 (22485206.75) ○
	50%	COOPT	1.00 (40031.86) ●	4.00 (243774.10) ○	2.00 (249802.00) ●	3.00 (232.00) ○	2.00 (7260872.48) ●
		DSA	2.10 (40744.82) ●	2.00 (12297.39) ●	1.00 (245000.00) ●	2.00 (230.00) ●	1.00 (5986933.68) ●
		MGM	3.10 (41336.99) ○	1.00 (10064.70) ●	3.00 (499800.00) ○	1.00 (224.00) ●	3.00 (16790187.12) ○
		MGM-2	3.80 (41580.80) ○	3.00 (25416.12) ○	4.00 (721673.95) ○	4.00 (817.00) ○	4.00 (22506616.02) ○
	90%	COOPT	1.00 (88885.45) ●	4.00 (243774.10) ○	2.00 (249802.00) ●	3.00 (232.00) ○	2.00 (7254039.92) ●
		DSA	2.00 (89712.99) ●	2.00 (12288.63) ●	1.00 (245000.00) ●	2.00 (230.00) ●	1.00 (5987614.78) ●
		MGM	3.30 (90420.09) ○	1.00 (10068.59) ●	3.00 (499800.00) ○	1.00 (224.00) ●	3.00 (16790398.80) ○
		MGM-2	3.70 (90510.93) ○	3.00 (25398.59) ○	4.00 (721730.34) ○	4.00 (817.00) ○	4.00 (22520620.59) ○
90%	30%	COOPT	1.10 (42312.63) ●	4.00 (442866.20) ○	2.00 (445802.00) ●	3.00 (232.00) ○	2.00 (13024021.84) ●
		DSA	1.90 (43030.38) ●	2.00 (22008.49) ●	1.00 (445000.00) ●	2.00 (230.00) ●	1.00 (10876939.20) ●
		MGM	3.10 (45616.36) ○	1.00 (18184.27) ●	3.00 (907800.00) ○	1.00 (224.00) ●	3.00 (30484287.12) ○
		MGM-2	3.90 (46082.85) ○	3.00 (46473.28) ○	4.00 (1310240.11) ○	4.00 (817.00) ○	4.00 (40708780.98) ○
	50%	COOPT	1.00 (82253.40) ●	4.00 (443307.10) ○	2.00 (445802.00) ●	3.00 (232.00) ○	2.00 (13022035.36) ●
		DSA	2.00 (83282.73) ●	2.00 (22007.21) ●	1.00 (445000.00) ●	2.00 (230.00) ●	1.00 (10875149.41) ●
		MGM	3.10 (86322.02) ○	1.00 (18184.98) ●	3.00 (907800.00) ○	1.00 (224.00) ●	3.00 (30484443.76) ○
		MGM-2	3.90 (86625.48) ○	3.00 (46528.25) ○	4.00 (1310253.86) ○	4.00 (817.00) ○	4.00 (40716952.89) ○
	90%	COOPT	1.00 (171492.33) ●	4.00 (443307.10) ○	2.00 (445802.00) ●	3.00 (232.00) ○	2.00 (13012964.48) ●
		DSA	2.00 (172477.48) ●	2.00 (22006.11) ●	1.00 (445000.00) ●	2.00 (230.00) ●	1.00 (10874772.05) ●
		MGM	3.10 (175464.34) ○	1.00 (18177.06) ●	3.00 (907800.00) ○	1.00 (224.00) ●	3.00 (30484102.00) ○
		MGM-2	3.90 (175901.36) ○	3.00 (46546.53) ○	4.00 (1310311.37) ○	4.00 (817.00) ○	4.00 (40732696.34) ○

Tabela 7.6: Resultados dos experimentos envolvendo problemas com 100 variáveis em redes aleatórias (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (9054.95) ●	4.00 (143799.10) ○	2.00 (155232.00) ●	3.00 (232.00) ○	2.00 (4482489.92) ●
		DSA	2.00 (9630.02) ●	2.00 (10122.93) ●	1.00 (148500.00) ●	2.00 (230.00) ●	1.00 (3627377.59) ●
		MGM	3.70 (10144.12) ○	1.00 (8405.33) ●	3.00 (302940.00) ○	1.00 (224.00) ●	3.00 (10175962.64) ○
		MGM-2	3.30 (10036.21) ○	3.00 (18206.42) ○	4.00 (437584.87) ○	4.00 (817.00) ○	4.00 (13682162.87) ○
	50%	COOPT	1.00 (20979.74) ●	4.00 (144823.00) ○	2.00 (155232.00) ●	3.00 (232.00) ○	2.00 (4478856.04) ●
		DSA	2.00 (21666.51) ●	2.00 (10324.44) ●	1.00 (148500.00) ●	2.00 (230.00) ●	1.00 (3627411.74) ●
		MGM	3.60 (22106.30) ○	1.00 (8564.98) ●	3.00 (302940.00) ○	1.00 (224.00) ●	3.00 (10176732.08) ○
		MGM-2	3.40 (22069.89) ○	3.00 (18496.27) ○	4.00 (437660.70) ○	4.00 (817.00) ○	4.00 (13702189.42) ○
	90%	COOPT	1.00 (50101.86) ●	4.00 (145194.50) ○	2.00 (155232.00) ●	3.00 (232.00) ○	2.00 (4475316.50) ●
		DSA	2.10 (50780.11) ●	2.00 (10279.88) ●	1.00 (148500.00) ●	2.00 (230.00) ●	1.00 (3627657.48) ●
		MGM	3.80 (51354.88) ○	1.00 (8529.47) ●	3.00 (302940.00) ○	1.00 (224.00) ●	3.00 (10177223.16) ○
		MGM-2	3.10 (51228.21) ○	3.00 (18435.46) ○	4.00 (437753.26) ○	4.00 (817.00) ○	4.00 (13721454.36) ○
50%	30%	COOPT	1.00 (19756.09) ●	4.00 (246040.60) ○	2.00 (252252.00) ●	3.00 (232.00) ○	2.00 (7336614.16) ●
		DSA	2.00 (20301.02) ●	2.00 (15208.88) ●	1.00 (247500.00) ●	2.00 (230.00) ●	1.00 (6046870.04) ●
		MGM	3.40 (20822.98) ○	1.00 (12636.53) ●	3.00 (504900.00) ○	1.00 (224.00) ●	3.00 (16960180.48) ○
		MGM-2	3.60 (20864.60) ○	3.00 (28053.56) ○	4.00 (728971.49) ○	4.00 (817.00) ○	4.00 (22717529.21) ○
	50%	COOPT	1.00 (40768.07) ●	4.00 (243816.30) ○	2.00 (252252.00) ●	3.00 (232.00) ○	2.00 (7332078.40) ●
		DSA	2.00 (41435.85) ●	2.00 (15503.65) ●	1.00 (247500.00) ●	2.00 (230.00) ●	1.00 (6048197.75) ●
		MGM	3.40 (42340.94) ○	1.00 (12870.01) ●	3.00 (504900.00) ○	1.00 (224.00) ●	3.00 (16960819.36) ○
		MGM-2	3.60 (42283.89) ○	3.00 (28616.20) ○	4.00 (729036.29) ○	4.00 (817.00) ○	4.00 (22735893.66) ○
	90%	COOPT	1.00 (89973.29) ●	4.00 (244076.30) ○	2.00 (252252.00) ●	3.00 (232.00) ○	2.00 (7326756.06) ●
		DSA	2.00 (90819.94) ●	2.00 (15068.11) ●	1.00 (247500.00) ●	2.00 (230.00) ●	1.00 (6047444.96) ●
		MGM	3.80 (91598.52) ○	1.00 (12515.51) ●	3.00 (504900.00) ○	1.00 (224.00) ●	3.00 (16960963.32) ○
		MGM-2	3.20 (91510.16) ○	3.00 (28059.30) ○	4.00 (729082.77) ○	4.00 (817.00) ○	4.00 (22746910.82) ○
90%	30%	COOPT	1.00 (42966.62) ●	4.00 (442695.24) ○	2.00 (446292.00) ●	3.00 (232.00) ○	2.00 (13042460.00) ●
		DSA	2.00 (43612.95) ●	2.00 (23524.03) ●	1.00 (445500.00) ●	2.00 (230.00) ●	1.00 (10886328.60) ●
		MGM	3.00 (45856.03) ○	1.00 (19484.86) ●	3.00 (908820.00) ○	1.00 (224.00) ●	3.00 (30518473.92) ○
		MGM-2	4.00 (46414.00) ○	3.00 (47340.38) ○	4.00 (1311731.40) ○	4.00 (817.00) ○	4.00 (40763402.50) ○
	50%	COOPT	1.00 (82867.58) ●	4.00 (444404.20) ○	2.00 (446292.00) ●	3.00 (232.00) ○	2.00 (13039138.90) ●
		DSA	2.00 (83706.54) ●	2.00 (23576.64) ●	1.00 (445500.00) ●	2.00 (230.00) ●	1.00 (10888340.17) ●
		MGM	3.10 (86403.55) ○	1.00 (19524.84) ●	3.00 (908820.00) ○	1.00 (224.00) ●	3.00 (30518613.40) ○
		MGM-2	3.90 (86979.07) ○	3.00 (47159.62) ○	4.00 (1311763.25) ○	4.00 (817.00) ○	4.00 (40774858.85) ○
	90%	COOPT	1.00 (171584.99) ●	4.00 (442835.80) ○	2.00 (446292.00) ●	3.00 (232.00) ○	2.00 (13027626.74) ●
		DSA	2.00 (172543.35) ●	2.00 (23603.32) ●	1.00 (445500.00) ●	2.00 (230.00) ●	1.00 (10887890.84) ●
		MGM	3.00 (175394.29) ○	1.00 (19545.48) ●	3.00 (908820.00) ○	1.00 (224.00) ●	3.00 (30518815.88) ○
		MGM-2	4.00 (176118.91) ○	3.00 (47097.17) ○	4.00 (1311769.03) ○	4.00 (817.00) ○	4.00 (40774974.16) ○

Tabela 7.7: Resultados dos experimentos envolvendo problemas com 100 variáveis em redes de mundos pequenos (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (8357.02) ●	4.00 (137346.40) ○	2.00 (146902.00) ●	3.00 (232.00) ○	2.00 (4236781.30) ●
		DSA	2.00 (8893.56) ●	2.00 (8774.04) ●	1.00 (140000.00) ●	2.00 (230.00) ●	1.00 (3419814.33) ●
		MGM	3.60 (9295.69) ○	1.00 (7294.30) ●	3.00 (285600.00) ○	1.00 (224.00) ●	3.00 (9593399.88) ○
		MGM-2	3.40 (9225.66) ○	3.00 (15985.80) ○	4.00 (412581.08) ○	4.00 (817.00) ○	4.00 (12910811.17) ○
	50%	COOPT	1.00 (19152.95) ●	4.00 (136141.10) ○	2.00 (146902.00) ●	3.00 (232.00) ○	2.00 (4233463.80) ●
		DSA	2.00 (19857.38) ●	2.00 (8900.79) ●	1.00 (140000.00) ●	2.00 (230.00) ●	1.00 (3420297.17) ●
		MGM	3.40 (20332.50) ○	1.00 (7399.10) ●	3.00 (285600.00) ○	1.00 (224.00) ●	3.00 (9594488.16) ○
		MGM-2	3.60 (20360.63) ○	3.00 (16197.75) ○	4.00 (412628.94) ○	4.00 (817.00) ○	4.00 (12921492.59) ○
	90%	COOPT	1.00 (46877.69) ●	4.00 (137510.50) ○	2.00 (146902.00) ●	3.00 (232.00) ○	2.00 (4231101.78) ●
		DSA	2.00 (47595.16) ●	2.00 (8775.91) ●	1.00 (140000.00) ●	2.00 (230.00) ●	1.00 (3420407.31) ●
		MGM	3.70 (48089.78) ○	1.00 (7294.95) ●	3.00 (285600.00) ○	1.00 (224.00) ●	3.00 (9594703.16) ○
		MGM-2	3.30 (48086.11) ○	3.00 (16000.58) ○	4.00 (412718.29) ○	4.00 (817.00) ○	4.00 (12939511.94) ○
50%	30%	COOPT	1.00 (18654.56) ●	4.00 (238140.60) ○	2.00 (244902.00) ●	3.00 (232.00) ○	2.00 (7118584.84) ●
		DSA	2.00 (19325.27) ●	2.00 (14276.02) ●	1.00 (240000.00) ●	2.00 (230.00) ●	1.00 (5863634.09) ●
		MGM	3.30 (19891.86) ○	1.00 (11859.48) ●	3.00 (489600.00) ○	1.00 (224.00) ●	3.00 (16446909.52) ○
		MGM-2	3.70 (19942.25) ○	3.00 (26467.03) ○	4.00 (706933.47) ○	4.00 (817.00) ○	4.00 (22044464.39) ○
	50%	COOPT	1.00 (38848.10) ●	4.00 (237833.90) ○	2.00 (244902.00) ●	3.00 (232.00) ○	2.00 (7113947.02) ●
		DSA	2.00 (39506.65) ●	2.00 (14331.66) ●	1.00 (240000.00) ●	2.00 (230.00) ●	1.00 (5863969.50) ●
		MGM	3.40 (40336.75) ○	1.00 (11903.57) ●	3.00 (489600.00) ○	1.00 (224.00) ●	3.00 (16447466.76) ○
		MGM-2	3.60 (40435.82) ○	3.00 (26693.15) ○	4.00 (706971.14) ○	4.00 (817.00) ○	4.00 (22054752.61) ○
	90%	COOPT	1.00 (86844.60) ●	4.00 (237769.20) ○	2.00 (244902.00) ●	3.00 (232.00) ○	2.00 (7110766.18) ●
		DSA	2.00 (87675.31) ●	2.00 (14448.57) ●	1.00 (240000.00) ●	2.00 (230.00) ●	1.00 (5864536.21) ●
		MGM	3.60 (88511.11) ○	1.00 (11983.66) ●	3.00 (489600.00) ○	1.00 (224.00) ●	3.00 (16447070.92) ○
		MGM-2	3.40 (88426.71) ○	3.00 (26851.05) ○	4.00 (707013.04) ○	4.00 (817.00) ○	4.00 (22066722.35) ○
90%	30%	COOPT	1.00 (41786.84) ●	4.00 (438279.00) ○	2.00 (440902.00) ●	3.00 (232.00) ○	2.00 (12880473.62) ●
		DSA	2.00 (42638.69) ●	2.00 (23483.67) ●	1.00 (440000.00) ●	2.00 (230.00) ●	1.00 (10754463.55) ●
		MGM	3.10 (44726.58) ○	1.00 (19442.29) ●	3.00 (897600.00) ○	1.00 (224.00) ●	3.00 (30142447.80) ○
		MGM-2	3.90 (45171.94) ○	3.00 (46864.81) ○	4.00 (1295529.25) ○	4.00 (817.00) ○	4.00 (40256625.00) ○
	50%	COOPT	1.00 (81477.16) ●	4.00 (437511.00) ○	2.00 (440902.00) ●	3.00 (232.00) ○	2.00 (12878478.12) ●
		DSA	2.00 (82442.05) ●	2.00 (23661.40) ●	1.00 (440000.00) ●	2.00 (230.00) ●	1.00 (10753960.87) ●
		MGM	3.00 (85137.46) ○	1.00 (19602.61) ●	3.00 (897600.00) ○	1.00 (224.00) ●	3.00 (30142467.28) ○
		MGM-2	4.00 (85934.55) ○	3.00 (47014.81) ○	4.00 (1295580.96) ○	4.00 (817.00) ○	4.00 (40273116.26) ○
	90%	COOPT	1.00 (169497.03) ●	4.00 (438294.80) ○	2.00 (440902.00) ●	3.00 (232.00) ○	2.00 (12870407.92) ●
		DSA	2.00 (170335.47) ●	2.00 (23685.99) ●	1.00 (440000.00) ●	2.00 (230.00) ●	1.00 (10753523.69) ●
		MGM	3.20 (173210.38) ○	1.00 (19624.34) ●	3.00 (897600.00) ○	1.00 (224.00) ●	3.00 (30142310.80) ○
		MGM-2	3.80 (173501.49) ○	3.00 (46893.86) ○	4.00 (1295561.78) ○	4.00 (817.00) ○	4.00 (40265458.56) ○

Tabela 7.8: Resultados dos experimentos envolvendo problemas com 100 variáveis em redes livres de escala (CD: 1.48)

Densidade	Tightness	Algoritmo	Custo da solução	NCCC	Número de mensagens	Tamanho da maior mensagem	Throughput de rede
30%	30%	COOPT	1.00 (8017.22) ●	4.00 (96094.40) ○	2.00 (136612.00) ●	3.00 (232.00) ○	2.00 (3929620.06) ●
		DSA	2.10 (8549.56) ●	2.00 (21469.23) ●	1.00 (129500.00) ●	2.00 (230.00) ●	1.00 (3158007.30) ●
		MGM	3.10 (8780.63) ○	1.00 (17813.82) ●	3.00 (264180.00) ○	1.00 (224.00) ●	3.00 (8871474.20) ○
		MGM-2	3.80 (8980.10) ○	3.00 (42143.33) ○	4.00 (381776.13) ○	4.00 (817.00) ○	4.00 (11968909.41) ○
	50%	COOPT	1.00 (18355.04) ●	4.00 (96037.30) ○	2.00 (136612.00) ●	3.00 (232.00) ○	2.00 (3928111.06) ●
		DSA	2.00 (19033.50) ●	2.00 (21519.98) ●	1.00 (129500.00) ●	2.00 (230.00) ●	1.00 (3157556.12) ●
		MGM	3.40 (19377.74) ○	1.00 (17855.76) ●	3.00 (264180.00) ○	1.00 (224.00) ●	3.00 (8871556.52) ○
		MGM-2	3.60 (19493.79) ○	3.00 (42058.89) ○	4.00 (381868.15) ○	4.00 (817.00) ○	4.00 (11980710.37) ○
	90%	COOPT	1.00 (43985.29) ●	4.00 (96946.80) ○	2.00 (136612.00) ●	3.00 (232.00) ○	2.00 (3925657.92) ●
		DSA	2.00 (44756.45) ●	2.00 (21203.11) ●	1.00 (129500.00) ●	2.00 (230.00) ●	1.00 (3158132.51) ●
		MGM	3.30 (45300.36) ○	1.00 (17586.32) ●	3.00 (264180.00) ○	1.00 (224.00) ●	3.00 (8872007.12) ○
		MGM-2	3.70 (45309.90) ○	3.00 (40696.59) ○	4.00 (382033.54) ○	4.00 (817.00) ○	4.00 (12011482.82) ○
50%	30%	COOPT	1.00 (16151.02) ●	4.00 (164792.70) ○	2.00 (215502.00) ●	3.00 (232.00) ○	2.00 (6250235.10) ●
		DSA	2.00 (16866.39) ●	2.00 (23091.41) ●	1.00 (210000.00) ●	2.00 (230.00) ●	1.00 (5125294.75) ●
		MGM	3.60 (17368.00) ○	1.00 (19151.56) ●	3.00 (428400.00) ○	1.00 (224.00) ●	3.00 (14388021.00) ○
		MGM-2	3.40 (17315.17) ○	3.00 (46130.31) ●	4.00 (618643.54) ○	4.00 (817.00) ○	4.00 (19312228.78) ○
	50%	COOPT	1.00 (33739.84) ●	4.00 (165764.00) ○	2.00 (215502.00) ●	3.00 (232.00) ○	2.00 (6247849.96) ●
		DSA	2.00 (34537.01) ●	2.00 (23309.64) ●	1.00 (210000.00) ●	2.00 (230.00) ●	1.00 (5125488.41) ●
		MGM	3.50 (35167.68) ○	1.00 (19335.24) ●	3.00 (428400.00) ○	1.00 (224.00) ●	3.00 (14388117.12) ○
		MGM-2	3.50 (35150.49) ○	3.00 (46008.48) ○	4.00 (618734.87) ○	4.00 (817.00) ○	4.00 (19337343.71) ○
	90%	COOPT	1.00 (75934.54) ●	4.00 (163990.60) ○	2.00 (215502.00) ●	3.00 (232.00) ○	2.00 (6244658.14) ●
		DSA	2.00 (76815.30) ●	2.00 (23361.33) ●	1.00 (210000.00) ●	2.00 (230.00) ●	1.00 (5124945.87) ●
		MGM	3.50 (77407.10) ○	1.00 (19378.12) ●	3.00 (428400.00) ○	1.00 (224.00) ●	3.00 (14388323.32) ○
		MGM-2	3.50 (77377.19) ○	3.00 (45913.93) ○	4.00 (618800.13) ○	4.00 (817.00) ○	4.00 (19346613.88) ○
90%	30%	COOPT	1.00 (30469.57) ●	4.00 (279217.80) ○	2.00 (343882.00) ●	3.00 (232.00) ○	2.00 (10027392.94) ●
		DSA	2.00 (31096.82) ●	2.00 (24345.52) ●	1.00 (341000.00) ●	2.00 (230.00) ●	1.00 (8327481.15) ●
		MGM	3.00 (31895.84) ○	1.00 (20193.88) ●	3.00 (695640.00) ○	1.00 (224.00) ●	3.00 (23365431.72) ○
		MGM-2	4.00 (32384.19) ○	3.00 (48927.61) ○	4.00 (1004221.73) ○	4.00 (817.00) ○	4.00 (31262308.44) ○
	50%	COOPT	1.00 (60503.23) ●	4.00 (279892.10) ○	2.00 (343882.00) ●	3.00 (232.00) ○	2.00 (10024679.70) ●
		DSA	2.00 (61481.29) ●	2.00 (24371.40) ●	1.00 (341000.00) ●	2.00 (230.00) ●	1.00 (8328876.98) ●
		MGM	3.10 (62540.99) ○	1.00 (20213.21) ●	3.00 (695640.00) ○	1.00 (224.00) ●	3.00 (23365839.64) ○
		MGM-2	3.90 (62909.25) ○	3.00 (48880.26) ●	4.00 (1004257.38) ○	4.00 (817.00) ○	4.00 (31271679.66) ○
	90%	COOPT	1.00 (129142.93) ●	4.00 (285348.40) ○	2.00 (343882.00) ●	3.00 (232.00) ○	2.00 (10019659.76) ●
		DSA	2.00 (130193.51) ●	2.00 (24298.10) ●	1.00 (341000.00) ●	2.00 (230.00) ●	1.00 (8327884.68) ●
		MGM	3.20 (131273.82) ○	1.00 (20152.86) ●	3.00 (695640.00) ○	1.00 (224.00) ●	3.00 (23365758.96) ○
		MGM-2	3.80 (131718.21) ○	3.00 (48691.46) ○	4.00 (1004280.08) ○	4.00 (817.00) ○	4.00 (31277280.30) ○

## Referências Bibliográficas

- ACEBRÓN, J. A. et al. The kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.*, v. 77, p. 137–185, 2005.
- AGRAWAL, P.; KUMAR, A.; VARAKANTHAM, P. Near-optimal decentralized power supply restoration in smart grids. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015. (AAMAS '15), p. 1275–1283. ISBN 978-1-4503-3413-6.
- ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, American Physical Society, v. 74, p. 47–97, 2002.
- ALI, S. M.; KOENIG, S.; TAMBE, M. Preprocessing techniques for accelerating the dcop algorithm adopt. In: DIGNUM, F. et al. (Ed.). *AAMAS*. [S.l.]: ACM, 2005. p. 1041–1048. ISBN 1-59593-094-9.
- AMIGONI, F.; CASTELLETTI, A.; GIULIANI, M. Modeling the management of water resources systems using multi-objective dcops. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015. (AAMAS '15), p. 821–829. ISBN 978-1-4503-3413-6.
- ARENAS, A. et al. Synchronization in complex networks. *Phys. Rep.*, v. 469, p. 93–153, 2008.
- ARENAS, A.; DÍAZ-GUILERA, A.; PÉREZ-VICENTE, C. J. Synchronization processes in complex networks. *Physica D: Nonlinear Phenomena*, v. 224, n. 1, p. 27 – 34, 2006. ISSN 0167-2789. Dynamics on Complex Networks and Applications.
- BARABÁSI, A.-L. *Linked: The New Science of Networks*. 1st. ed. [S.l.]: Basic Books, 2002. Hardcover.
- BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, n. 5439, p. 509–512, 1999.

- BARAHONA, M.; PECORA, L. M. Synchronization in small-world systems. *Phys. Rev. Lett.*, v. 89, n. 5, p. 054101, 2002.
- BARTÁK, R. Theory and practice of constraint propagation. In: *Proceedings of the 3rd Workshop on Constraint Programming in Decision and Control*. [S.l.: s.n.], 2001. p. 7–14.
- BOWRING, E.; TAMBE, M.; YOKOO, M. Multiply-constrained distributed constraint optimization. In: NAKASHIMA, H. et al. (Ed.). *AAMAS*. [S.l.]: ACM, 2006. p. 1413–1420. ISBN 1-59593-303-4.
- BUCK, J.; BUCK, E. Synchronous fireflies. *Scientific American*, v. 234, p. 74–85, 1976.
- CARPENTER, C. J. et al. Disaster evacuation support. In: *AAAI*. [S.l.]: AAAI Press, 2007. p. 1964–1965. ISBN 978-1-57735-323-2.
- CENCINI, M.; CECCONI, F.; VULPIANI, A. *Chaos: From Simple Models to Complex Systems*. [S.l.]: World Scientific, 2010. (Series on advances in statistical mechanics). ISBN 9789814277655.
- CERQUIDES, J.; PICARD, G.; RODRÍGUEZ-AGUILAR, J. A. Designing a marketplace for the trading and distribution of energy in the smart grid. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015. (AAMAS '15), p. 1285–1293. ISBN 978-1-4503-3413-6.
- CHECHETKA, A.; SYCARA, K. No-commitment branch and bound search for distributed constraint optimization. In: NAKASHIMA, H. et al. (Ed.). *AAMAS*. [S.l.]: ACM, 2006. p. 1427–1429. ISBN 1-59593-303-4.
- CHEN, Z. et al. Bd-adopt: a hybrid dcop algorithm with best-first and depth-first search strategies. *Artificial Intelligence Review*, Jan 2017. ISSN 1573-7462.
- DEGESYS, J. et al. Desync: self-organizing desynchronization and tdma on wireless sensor networks. In: *IPSN*. [S.l.]: ACM, 2007. p. 11–20.
- DÍAZ-GUILERA, A. et al. Synchronization in random geometric graphs. *Int. J. Bifurcat. Chaos*, v. 19, p. 687–693, 2009.
- DÖRFLER, F.; BULLO, F. On the critical coupling for kuramoto oscillators. *SIAM J. Applied Dynamical Systems*, v. 10, n. 3, p. 1070–1099, 2011.

- DÖRFLER, F.; BULLO, F. Exploring synchronization in complex oscillator networks. In: *CDC*. [S.l.]: IEEE, 2012. p. 7157–7170. ISBN 978-1-4673-2065-8.
- DORFLER, F.; BULLO, F. Synchronization and transient stability in power networks and nonuniform kuramoto oscillators. *SIAM Journal on Control and Optimization*, Society for Industrial and Applied Mathematics, v. 50, n. 3, p. 1616–1642, 2012.
- DÖRFLER, F.; M., C.; F., B. Synchronization in complex oscillator networks and smart grids. *Proceedings of the National Academy of Sciences*, v. 110, n. 6, p. 2005–2010, 2013.
- DYE, J. Ionic and synaptic mechanisms underlying a brainstem oscillator: an in vitro study of the pacemaker nucleus of apteronotus. *J Comp Physiol A*, v. 168, n. 5, p. 521–532, 1991.
- ERDÖS, P.; RÉNYI, A. On random graphs i. *Publicationes Mathematicae Debrecen*, v. 6, p. 290, 1959.
- FARINELLI, A. et al. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In: PADGHAM, L. et al. (Ed.). *AAMAS*. [S.l.]: IFAAMAS, 2008. p. 639–646. ISBN 978-0-9817381-1-6.
- FAVE, F. M. D. et al. Bounded decentralised coordination over multiple objectives. In: *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011. (AAMAS '11), p. 371–378. ISBN 0-9826571-5-3, 978-0-9826571-5-7.
- FERBER, J. *Multi-agent systems - an introduction to distributed artificial intelligence*. [S.l.]: Addison-Wesley-Longman, 1999.
- FIORETTO, F. et al. Gd-gibbs: A gpu-based sampling algorithm for solving distributed constraint optimization problems. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014. (AAMAS '14), p. 1339–1340. ISBN 978-1-4503-2738-1.
- FISCHER, M. J. The consensus problem in unreliable distributed systems (a brief survey). In: KARPINSKI, M. (Ed.). *FCT*. [S.l.]: Springer, 1983. (Lecture Notes in Computer Science, v. 158), p. 127–140. ISBN 3-540-12689-9.
- FORTUNATO, S.; CASTELLANO, C. Community structure in graphs. In: MEYERS, R. A. (Ed.). *Computational Complexity*. [S.l.]: Springer New York, 2012. p. 490–512. ISBN 978-1-4614-1799-6.

- FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 11, n. 1, p. 86–92, 1940. ISSN 00034851.
- GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. First edition. [S.l.]: W. H. Freeman, 1979. ISBN 0716710455.
- GASTON, M. E.; DESJARDINS, M. Agent-organized networks for multi-agent production and exchange. In: *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*. [S.l.]: AAAI Press, 2005. (AAAI'05), p. 77–82. ISBN 1-57735-236-x.
- GENT, I. P. et al. Solving quantified constraint satisfaction problems. *Artif. Intell.*, v. 172, n. 6-7, p. 738–771, 2008.
- GERSHMAN, A.; MEISELS, A.; ZIVAN, R. Asynchronous forward bounding for distributed cops. *J. Artif. Intell. Res. (JAIR)*, v. 34, p. 61–88, 2009.
- GHOSH, A. K.; CHANCE, B.; PYE, E. K. Metabolic coupling and synchronization of nadh oscillations in yeast cell populations. *Arch Biochem Biophys*, v. 145, n. 1, p. 319–31, 1971. ISSN 0003-9861.
- GHOSH, S.; KUMAR, A.; VARAKANTHAM, P. Probabilistic inference based message-passing for resource constrained dcops. In: YANG, Q.; WOOLDRIDGE, M. (Ed.). *IJCAI*. [S.l.]: AAAI Press, 2015. p. 411–417. ISBN 978-1-57735-738-4.
- GRANOVETTER, M. The strength of weak ties. *The American Journal of Sociology*, v. 78, n. 6, p. 1360–1380, maio 1973.
- GREENSTADT, R. et al. Experimental analysis of privacy loss in dcop algorithms. In: NAKASHIMA, H. et al. (Ed.). *AAMAS*. [S.l.]: ACM, 2006. p. 1424–1426. ISBN 1-59593-303-4.
- GRINSHPOUN, T. et al. Asymmetric distributed constraint optimization problems. *J. Artif. Intell. Res. (JAIR)*, v. 47, p. 613–647, 2013.
- GROSS, T.; BLASIUS, B. Adaptive coevolutionary networks: a review. *J. R. Soc. Interface*, The Royal Society, v. 5, n. 20, p. 259–271, 2008.
- GRUBSHTEIN, A. et al. Local search for distributed asymmetric optimization. In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent*

*System*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2010. (AAMAS '10), p. 1015–1022. ISBN 978-0-9826571-1-9.

GUSHCHIN, A.; MALLADA, E.; TANG, A. Synchronization of phase-coupled oscillators with plastic coupling strength. In: *ITA*. [S.l.]: IEEE, 2015. p. 291–300. ISBN 978-1-4799-7195-4.

HIRAYAMA, K.; YOKOO, M. Distributed partial constraint satisfaction problem. In: *Principles and Practice of Constraint Programming*. [S.l.: s.n.], 1997. p. 222–236.

HOANG, K. D. et al. Proactive dynamic distributed constraint optimization. In: *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2016. (AAMAS '16), p. 597–605. ISBN 978-1-4503-4239-1.

HONG, H.; STROGATZ, S. H. Kuramoto Model of Coupled Oscillators with Positive and Negative Coupling Parameters: An Example of Conformist and Contrarian Oscillators. *Physical Review Letters*, American Physical Society, v. 106, n. 5, p. 102–104, fev. 2011. Disponível em: <<http://dx.doi.org/10.1103/physrevlett.106.054102>>.

ITO, J.; KANEKO, K. Spontaneous structure formation in a network of chaotic units with variable connection strengths. *Phys. Rev. Lett.*, American Physical Society, v. 88, n. 2, p. 028701, 2001.

JENNINGS, N. R.; BUSSMANN, S. Agent-based control systems. *IEEE Control Systems Magazine*, v. 23, p. 61–74, 2003.

JENNINGS, N. R.; SYCARA, K.; WOOLDRIDGE, M. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, Springer Netherlands, v. 1, p. 7–38, 1998. ISSN 1387-2532.

JUNGES, R.; BAZZAN, A. L. C. Evaluating the performance of dcop algorithms in a real world, dynamic problem. In: PADGHAM, L. et al. (Ed.). *AAMAS*. [S.l.]: IFAAMAS, 2008. p. 599–606. ISBN 978-0-9817381-1-6.

KIEKINTVELD, C. et al. Asynchronous algorithms for approximate distributed constraint optimization with quality bounds. In: HOEK, W. van der et al. (Ed.). *AAMAS*. [S.l.]: IFAAMAS, 2010. p. 133–140. ISBN 978-0-9826571-1-9.

KIM, Y.; LESSER, V. R. Djao: A communication-constrained dcop algorithm that combines features of adopt and action-gdl. In: BRODLEY, C. E.; STONE, P. (Ed.). *AAAI*. [S.l.]: AAAI Press, 2014. p. 2680–2687. ISBN 978-1-57735-661-5.

KLEIN, D. J.; LALISH, E.; MORGANSEN, K. A. On controlled sinusoidal phase coupling. In: *Proceedings of the 2009 Conference on American Control Conference*. [S.l.]: IEEE Press, 2009. p. 616–622.

KURAMOTO, Y. Self-entrainment of a population of coupled nonlinear oscillators. In: ARAKI, H. (Ed.). *International Symposium on Mathematical Problems in Theoretical Physics, Lecture Notes in Physics, Vol. 39*,. New York, NY, USA: Springer, 1975. p. 420–422.

KURAMOTO, Y. *Chemical oscillations, waves, and turbulence*. New York, NY, USA: Springer-Verlag, 1984.

LE, T. et al. Er-dcops: A framework for distributed constraint optimization with uncertainty in constraint utilities. In: *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2016. (AAMAS '16), p. 606–614. ISBN 978-1-4503-4239-1.

LE, T. et al. Solving distributed constraint optimization problems using logic programming. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2015. (AAAI'15), p. 1174–1181. ISBN 0-262-51129-0.

LÉAUTÉ, T.; OTTENS, B.; SZYMANEK, R. Frodo 2.0: An open-source framework for distributed constraint optimization. In: *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*. Pasadena, California, USA: AAAI Press, 2009. p. 160–164.

LEITE, A. R. et al. Improving the distributed constraint optimization using social network analysis. In: *Proceedings of the 20th Brazilian Conference on Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2010. (SBIA'10), p. 243–252. ISBN 3-642-16137-5, 978-3-642-16137-7.

LEITE, A. R.; ENEMBRECK, F.; BARTHÈS, J.-P. A. Distributed constraint optimization problems: Review and perspectives. *Expert Systems with Applications*, v. 41, n. 11, p. 5139–5157, 2014. ISSN 0957-4174.

LEITE, A. R.; GIACOMET, B.; ENEMBRECK, F. Railroad driving model based on distributed constraint optimization. In: *IAT*. [S.l.]: IEEE, 2009. p. 474–481.

LESSER, V.; ORTIZ, C.; TAMBE, M. (Ed.). *Distributed Sensor Networks: A Multiagent Perspective*. [S.l.]: Kluwer Academic Publishers, 2003.

- LUENBERGER, D. G. *Introduction to dynamic systems : theory, models, and applications*. New York, Chichester, Brisbane: J. Wiley & Sons, 1979.
- MAHESWARAN, R. T. et al. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, Hingham, MA, USA, v. 13, n. 1, p. 27–60, jul. 2006. ISSN 1387-2532.
- MAHESWARAN, R. T.; PEARCE, J. P.; TAMBE, M. Distributed algorithms for dcop: A graphical-game-based approach. In: BADER, D. A.; KHOKHAR, A. A. (Ed.). *ISCA PDCS*. [S.l.]: ISCA, 2004. p. 432–439. ISBN 1-880843-52-8.
- MAHESWARAN, R. T. et al. Taking dcop to the real world: Efficient complete solutions for distributed multi-event scheduling. In: *AAMAS*. [S.l.]: IEEE Computer Society, 2004. p. 310–317. ISBN 1-58113-864-4.
- MAILLER, R.; LESSER, V. R. Solving distributed constraint optimization problems using cooperative mediation. In: *AAMAS*. [S.l.]: IEEE Computer Society, 2004. p. 438–445. ISBN 1-58113-864-4.
- MATSUI, T. et al. Resource constrained distributed constraint optimization with virtual variables. In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*. [S.l.]: AAAI Press, 2008. (AAAI'08), p. 120–125. ISBN 978-1-57735-368-3.
- MATSUI, T. et al. A quantified distributed constraint optimization problem. In: HOEK, W. van der et al. (Ed.). *AAMAS*. [S.l.]: IFAAMAS, 2010. p. 1023–1030. ISBN 978-0-9826571-1-9.
- MEIER, M.; HASCHKE, R.; RITTER, H. J. Perceptual grouping through competition in coupled oscillator networks. *Neurocomputing*, v. 141, p. 76–83, 2014.
- MEISELS, A. et al. Comparing performance of distributed constraints processing algorithms. In: *AAMAS - Workshop on Distributed Constraint Reasoning DCR*. [S.l.]: AAAI Press, 2002. p. 86–93.
- MILGRAM, S. The small world problem. *Psychology Today*, v. 67, n. 1, p. 61–67, 1967.
- MODI, P. J. et al. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, v. 161, n. 1-2, p. 149–180, 2005.
- MONTEIRO, T. L. et al. A multi-agent approach to optimal channel assignment in wlans. In: *WCNC*. [S.l.]: IEEE, 2012. p. 2637–2642. ISBN 978-1-4673-0436-8.

- MULDOON, C. et al. Distributed constraint optimisation for resource limited sensor networks. *Sci. Comput. Program.*, v. 78, n. 5, p. 583–593, 2013.
- NEMENYI, P. *Distribution-free multiple comparisons*. Tese (Doutorado) — Princeton University, New Jersey, USA, 1963.
- NETZER, A.; GRUBSHTEIN, A.; MEISELS, A. Concurrent forward bounding for distributed constraint optimization problems. *Artif. Intell.*, v. 193, p. 186–216, 2012.
- NGUYEN, D. T.; YEOH, W.; LAU, H. C. Stochastic dominance in stochastic dcops for risk-sensitive applications. In: *AAMAS*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2012. p. 257–264. ISBN 0-9817381-1-7, 978-0-9817381-1-6.
- NGUYEN, D. T.; YEOH, W.; LAU, H. C. Distributed gibbs: A memory-bounded sampling-based dcop algorithm. In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013. (AAMAS), p. 167–174. ISBN 978-1-4503-1993-5.
- O’HARE, G. M. P.; JENNINGS, N. R. *Foundations of distributed artificial intelligence*. [S.l.]: Wiley, 1996. 1-576 p. (Sixth-generation computer technology series).
- OKIMOTO, T. et al. Lp-norm based algorithm for multi-objective distributed constraint optimization. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014. (AAMAS ’14), p. 1427–1428. ISBN 978-1-4503-2738-1.
- OLFATI-SABER, R.; MURRAY, R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, IEEE Computer Society, v. 49, n. 9, p. 1520–1533, 2004.
- OTTENS, B.; DIMITRAKAKIS, C.; FALTINGS, B. Duct: An upper confidence bound approach to distributed constraint optimization problems. In: *AAAI. Proceedings of the 26th conference of the AAAI*. [S.l.]: AAAI Press, 2012. p. 528–533.
- PAGLIARI, R.; SCAGLIONE, A. Scalable network synchronization with pulse-coupled oscillators. *IEEE Transactions on Mobile Computing*, IEEE Computer Society, v. 10, n. 3, p. 392–405, 2011. ISSN 1536-1233.

- PALEY, D. et al. Oscillator models and collective motion: Spatial patterns in the dynamics of engineered and biological networks. *IEEE Control Systems Magazine*, v. 27, n. 4, p. 89–105, 2007.
- PEARCE, J. P.; TAMBE, M.; MAHESWARAN, R. T. Solving multiagent networks using distributed constraint optimization. *AI Magazine*, v. 29, n. 3, p. 47–62, 2008.
- PECORA, L. M.; CARROLL, T. L. Synchronization in chaotic systems. *Phys. Rev. Lett.*, v. 64, p. 821–824, 1990.
- PECORA, L. M.; CARROLL, T. L. Master stability functions for synchronized coupled systems. *Phys. Rev. Lett.*, v. 80, n. 10, p. 2109–2112, March 1998.
- PECORA, L. M. et al. Fundamentals of synchronization in chaotic systems, concepts, and applications. *Chaos*, v. 7, n. 4, p. 520–543, 1997.
- PERI, O.; MEISELS, A. Synchronizing for performance - dcop algorithms. In: FILIPE, J.; FRED, A. L. N. (Ed.). *ICAART*. [S.l.]: SciTePress, 2013. p. 5–14. ISBN 978-989-8565-38-9.
- PESKIN, C. S. *Mathematical aspects of heart physiology*. New York, NY, USA: Courant Institute of Mathematical Sciences, New York University, 1975.
- PETCU, A. *A Class of Algorithms for Distributed Constraint Optimization*. Tese (PhD. Thesis No. 3942) — Swiss Federal Institute of Technology (EPFL), Lausanne (Switzerland), 2007.
- PETCU, A.; FALTINGS, B. A scalable method for multiagent constraint optimization. In: KAELBLING, L. P.; SAFFIOTTI, A. (Ed.). *IJCAI*. [S.l.]: Professional Book Center, 2005. p. 266–271. ISBN 0938075934.
- PETCU, A.; FALTINGS, B. Superstabilizing, fault-containing distributed combinatorial optimization. In: *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*. [S.l.]: AAAI Press, 2005. (AAAI'05), p. 449–454. ISBN 1-57735-236-x.
- PETCU, A.; FALTINGS, B. Mb-dpop: A new memory-bounded algorithm for distributed optimization. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. (IJCAI'07), p. 1452–1457.

PETCU, A.; FALTINGS, B. Optimal solution stability in dynamic, distributed constraint optimization. In: *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. Washington, DC, USA: IEEE Computer Society, 2007. (IAT '07), p. 321–327. ISBN 0-7695-3027-3.

PETCU, A.; FALTINGS, B.; MAILLER, R. Pc-dpop: A new partial centralization algorithm for distributed optimization. In: VELOSO, M. M. (Ed.). *IJCAI*. [S.l.: s.n.], 2007. p. 167–172.

PIKOVSKY, A.; ROSENBLUM, M. G.; KURTHS, J. *Synchronization, A Universal Concept in Nonlinear Sciences*. Cambridge: Cambridge University Press, 2001.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2008. ISBN 3-900051-07-0.

ROGERS, A. et al. Bounded approximate decentralised coordination via the max-sum algorithm. *Artif. Intell.*, v. 175, n. 2, p. 730–759, 2011.

ROSSI, F.; BEEK, P. v.; WALSH, T. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. New York, NY, USA: Elsevier Science Inc., 2006. ISBN 0444527265.

RUST, P.; PICARD, G.; RAMPARANY, F. Using Message-passing DCOP Algorithms to Solve Energy-efficient Smart Environment Configuration Problems. In: KAMBHAMPATI, S. (Ed.). *International Joint Conference on Artificial Intelligence*. New York, United States: AAAI Press, 2016. p. 468–474.

SANDHOLM, T.; LESSER, V. R. Coalition formation among bounded rational agents. In: *IJCAI*. [S.l.]: Morgan Kaufmann, 1995. p. 662–671.

SCERRI, P.; VINCENT, R.; MAILLER, R. T. *Coordination of Large-Scale Multiagent Systems*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 1441938729, 9781441938725.

SEPULCHRE, R.; PALEY, D. A.; LEONARD, N. E. Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control*, v. 53, n. 3, p. 706–719, 2008.

SHERMAN, A.; RINZEL, J.; KEIZER, J. Emergence of organized bursting in clusters of pancreatic beta-cells by channel sharing. *Biophys J*, v. 54, n. 3, p. 411–425, 1988. ISSN 0006-3495.

- SHOHAM, Y.; LEYTON-BROWN, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, UK: Cambridge University Press, 2009. ISBN 978-0-521-89943-7.
- SILAGHI, M.-C.; YOKOO, M. Nogood based asynchronous distributed optimization (adopt ng). In: NAKASHIMA, H. et al. (Ed.). *AAMAS*. [S.l.]: ACM, 2006. p. 1389–1396. ISBN 1-59593-303-4.
- SIMEONE, O.; SPAGNOLINI, U. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP J. Wireless Comm. and Networking*, v. 2007, 2007.
- SIVRIKAYA, F.; YENER, B. Time synchronization in sensor networks: a survey. *IEEE Network*, v. 18, n. 4, p. 45–50, 2004.
- SKARDAL, P. S.; RESTREPO, J. G. Hierarchical synchrony of phase oscillators in modular networks. *Physical Review E, APS*, v. 85, n. 1, p. 016208, 2012.
- STROGATZ, S. H. Spontaneous synchronization in nature. In: *Frequency Control Symposium*. [S.l.]: IEEE, 1997. p. 2–4.
- STROGATZ, S. H. *Sync: The Emerging Science of Spontaneous Order*. [S.l.]: Hyperion, 2003. ISBN 9780786868445.
- SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- TANAKA, T.; AOYAGI, T. Optimal weighted networks of phase oscillators for synchronization. *Physical Review E*, v. 78, n. 4, p. 046210, 2008.
- TSANG, E. P. K. *Foundations of constraint satisfaction*. [S.l.]: Academic Press, 1993. I-XVIII, 1-421 p. (Computation in cognitive science). ISBN 978-0-12-701610-8.
- VINYALS, M. et al. Divide-and-coordinate: Dcops by agreement. In: HOEK, W. van der et al. (Ed.). *AAMAS*. [S.l.]: IFAAMAS, 2010. p. 149–156. ISBN 978-0-9826571-1-9.
- WALKER, T. J. Acoustic synchrony: Two mechanisms in the snowy tree cricket. *Science*, v. 166, n. 3907, p. 891–894, 1969.
- WASSERMAN, S.; FAUST, K. *Social network analysis: Methods and applications*. [S.l.]: Cambridge university press, 1994.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of small-world networks. *Nature*, v. 393, n. 6684, p. 440–442, 1998.

WEISS, G. (Ed.). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: MIT Press, 1999. ISBN 978-0-262-23203-6.

WINFREE, A. T. Biological rhythms and the behavior of populations of coupled oscillators. *J. Theor. Biol.*, v. 16, p. 15–42, 1967.

WOLFRAM, S. *The Mathematica book*. Fifth. Champaign, IL, USA: Wolfram Media, 2003. 1464 p. ISBN 1-57955-022-3.

WOOLDRIDGE, M. *An Introduction to Multiagent Systems*. 2. ed. Chichester, UK: Wiley, 2009. ISBN 978-0-470-51946-2.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, v. 10, p. 115–152, 1995.

WU, F.; JENNINGS, N. R. Regret-based multi-agent coordination with uncertain task rewards. In: *AAAI-14: Twenty-Eighth Conference on Artificial Intelligence*. [S.l.: s.n.], 2014. p. 1492–1499.

WU, J. et al. Clustering dynamics of nonlinear oscillator network: Application to graph coloring problem. *Physica D: Nonlinear Phenomena*, v. 240, n. 24, p. 1972 – 1978, 2011. ISSN 0167-2789.

YEDIDSION, H.; ZIVAN, R. Applying dco<sub>p</sub>\_mst to a team of mobile robots with directional sensing abilities. In: *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2016. (AAMAS '16), p. 1357–1358. ISBN 978-1-4503-4239-1.

YEOH, W. *Speeding up distributed constraint optimization search algorithms*. Tese (Doutorado) — University of Southern California, Los Angeles, CA, USA, 2010. AAI3434536.

YEOH, W.; FELNER, A.; KOENIG, S. Bnb-adopt: An asynchronous branch-and-bound dco<sub>p</sub> algorithm. *J. Artif. Intell. Res. (JAIR)*, IFAAMAS, Richland, SC, v. 38, p. 85–133, 2010.

YEOH, W. et al. Incremental dcop search algorithms for solving dynamic dcops. In: *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011. (AAMAS '11), p. 1069–1070. ISBN 0-9826571-7-X, 978-0-9826571-7-1.

YEOH, W.; YOKOO, M. Distributed problem solving. *AI Magazine*, v. 33, n. 3, p. 53–65, 2012.

YOKOO, M. et al. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 10, n. 5, p. 673–685, set. 1998. ISSN 1041-4347.

YOOK, S. H.; JEONG, H.; BARABÁSI, A. L. Weighted evolving networks. *Physical Review Letters*, v. 86, p. 5835–5838, 2001.

ZHANG, W. et al. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artif. Intell.*, v. 161, n. 1-2, p. 55–87, 2005.

ZHOU, C.; KURTHS, J. Dynamical weights and enhanced synchronization in adaptive complex networks. *Phys. Rev. Lett.*, v. 96, n. 16, p. 164102, 2006.

ZIVAN, R.; OKAMOTO, S.; PELED, H. Explorative anytime local search for distributed constraint optimization. *Artif. Intell.*, Elsevier Science Publishers Ltd., Essex, UK, v. 212, n. 1, p. 1–26, jul. 2014. ISSN 0004-3702.