

Topologias Dinâmicas para Modelo em Ilhas usando Evolução Diferencial

Rodolfo Ayala Lopes Costa

Orientadores:

Frederico Gadelha Guimarães

Felipe Campelo

Dissertação submetida à Escola de Engenharia
Universidade Federal de Minas Gerais
para obtenção do título de Mestre em Engenharia Elétrica

*Dedico este trabalho a Deus e minha família que sempre me incentivou e apoiou. Amo
muito vocês!*

Topologias Dinâmicas para Modelo em Ilhas usando Evolução Diferencial

Resumo

Nas últimas décadas, o desenvolvimento das metaheurísticas evolutivas tem proporcionado o aumento do domínio de aplicações, bem como, a resolução de problemas complexos. Em particular, o Modelo em Ilhas (IM - Island Model) é uma importante abordagem multipopulacional capaz de melhorar o desempenho destas metaheurísticas em problemas de alta complexidade. Apesar do seu sucesso, seu desempenho está diretamente relacionado à escolha adequada dos parâmetros deste modelo. Um dos parâmetro críticos para o IM é a topologia de migração. Ela é responsável por definir o fluxo migratório entre as subpopulações do modelo podendo afetar a qualidade da solução final e o ritmo de convergência. Como a maioria das topologias encontradas na literatura são estáticas e ingênuas, um novo conceito de fluxos migratórios dinâmicos, conhecido como topologia de migração dinâmica, tem sido estudado atualmente. Diversas topologias de migração dinâmicas para IM podem ser encontradas na literatura. Afim de compreender melhor os efeitos dessas abordagens no desempenho do IM utilizando o algoritmo de Evolução Diferencial (DE - Differential Evolution), este estudo apresenta uma análise experimental criteriosa das abordagens existentes. Além disso, uma análise das topologias obtidas pelas abordagens estudadas por meio de métricas de redes complexas também fornecem dicas sobre a interação entre as suas características e o desempenho do modelo. Os resultados demonstram que as topologias dinâmicas podem melhorar o desempenho do IM sem que haja perda significativa na qualidade das soluções finais.

Dynamic Topologies to Island Model using Differential Evolution

Abstract

Over the last decades, the development of evolutionary metaheuristics has proportionated the increase the range of applications, as well as, the solution of complex problems. In particular, the Island Model (IM) is an important multi-population approach able to improve the performance of these metaheuristics in high complex problems. Despite its success, its performance is closely related to the appropriate choice of the IM parameters. One of those parameters is the migration topology. It is responsible by defining the migratory flow between the subpopulations of the model, impacting the quality of solutions and convergence time. As the majority of the topologies found in the literature are static and naive, a new concept of dynamic migratory flows, known as dynamic migration topology, has been studied. Several dynamic migration topologies for IM can be found in the literature. In order to understand the effects of these approaches on the performance of the IM using the Differential Evolution (DE) algorithm, this study presents a thorough experimental analysis of the existing approaches. Furthermore, an analysis of the topologies obtained by the studied approaches by means of complex networks metrics also provides some insight on the interaction between their characteristics and the performance of the model. The results show that the dynamic migration topologies is able to improve the performance of the IM without significant loss in the final quality of solutions.

Agradecimentos

Agradeço a Deus por esta oportunidade e por suas bênçãos.

Agradeço aos meus orientadores Frederico Gadelha Guimarães e Felipe Campelo, pela confiança, orientação e ensinamentos. A disponibilidade e estímulo destes grandes mestres foram fundamentais para superação de mais esta etapa.

Agradeço a todos meus amigos e os companheiros do grupo informal POPES, Alan Freitas, Ricardo Sérgio e Rodrigo César pela amizade e trabalhos realizados. Ao Rodrigo César, um agradecimento especial pelo apoio, incentivo e colaboração.

Agradeço aos meus pais, Nem e Marise, e minha irmã Túlia por todo amor e confiança. O incentivo de vocês sempre me faz almejar grandes conquistas.

Agradeço aos meus familiares que me acompanharam e torceram por mim nesta caminhada. Aos meus afilhados, Pedro e Raul, por tornarem meus dias mais alegres.

Ao CNPq e à CAPES pelo apoio financeiro, sem o qual este trabalho não seria possível.

Sumário

Lista de Figuras	xv
Lista de Tabelas	xvii
Nomenclatura	1
1 Introdução	3
1.1 Introdução	3
1.2 Objetivos	6
1.2.1 Objetivos Gerais	6
1.2.2 Objetivos Específicos	6
1.3 Motivação	6
1.4 Trabalhos Relacionados	7
1.5 Organização do Trabalho	8
2 Fundamentos Teóricos	11
2.1 Otimização	11
2.2 Computação Natural e Algoritmos Evolutivos	13
2.3 Evolução Diferencial	15
2.3.1 Mutação Diferencial	17

2.3.2	Recombinação	18
2.3.3	Seleção	19
2.3.4	Pseudocódigo	19
3	Estratégias de Paralelização e o Modelo em Ilhas	21
3.1	Estratégias de Paralelização	21
3.2	Modelo em Ilhas	23
3.2.1	Número de Ilhas	27
3.2.2	Frequência de Migração	27
3.2.3	Taxa de Migração	28
3.2.4	Tipo de Sincronização	28
3.2.5	Política de Migração	29
3.2.6	Topologia de Migração	29
4	Topologias Dinâmicas para Modelo em Ilhas	31
4.1	Características	31
4.2	Topologias baseadas em Métodos de Seleção	32
4.2.1	Seleção Aleatória	32
4.2.2	Seleção por Roleta	34
4.2.3	Seleção por Torneio	36
4.3	Topologias baseadas em Métodos de Recompensa	37
4.3.1	Grafo Completo - Recompensa Simples	37
4.3.2	Sistema Multiagentes - Q-learning	39
4.3.3	Sistema Multiagentes - Upper Confidence Bound	43
4.3.4	Indicadores Populacionais	45
4.4	Topologias baseadas em Técnicas de Agrupamento	47

4.4.1	K-medoids	47
4.4.2	Métricas de distância	48
5	Resultados	49
5.1	Configuração dos Testes	49
5.1.1	Método de Otimização	49
5.1.2	Problemas de Otimização	50
5.1.3	Parâmetros - Modelo em Ilhas	50
5.1.4	Nomenclatura das Topologias Estudadas	50
5.1.5	Testes Estatísticos	51
5.2	Resultados Comparativos	52
5.2.1	Topologia Aleatória	52
5.2.2	Topologia Roleta	54
5.2.3	Topologia Torneio	56
5.2.4	Topologia Grafo Completo	57
5.2.5	Topologia Qlearning	59
5.2.6	Topologia UCB	62
5.2.7	Topologia K-medoids	65
5.2.8	Resultados Comparativos: Topologias Dinâmicas	67
5.3	Comportamento das Topologias	73
6	Conclusões	79
A	Apêndice	83
A.1	Funções de Teste	83
A.1.1	Função 1: Sphere Deslocada	83

A.1.2	Função 2: Schwefel Deslocada	83
A.1.3	Função 3: Rosenbrock Deslocada	84
A.1.4	Função 4: Rastrigin Deslocada	84
A.1.5	Função 5: Griewank Deslocada	85
A.1.6	Função 6: Ackley Deslocada	85
Referências Bibliográficas		87

Lista de Figuras

1.1	Diagrama representativo do fluxo básico dos Algoritmos Evolutivos. . . .	4
2.1	Representação gráfica de problemas de otimização bidimensionais.	12
2.2	Ilustração do procedimento de geração de uma solução mutante.	18
3.1	Projeto de paralelização do método de avaliação das soluções de um EA.	23
3.2	Processo de divisão da população total em subpopulações.	24
3.3	Ilustração da distribuição espacial das subpopulações para o problema de otimização Schwefel unidimensional.	25
3.4	Processo migratório para IM com 4 topologias usando topologia em anel.	26
3.5	Tradicionais topologias de migração estáticas.	30
4.1	Processo de definição da topologia de migração aleatória em um modelo composto por 4 ilhas.	33
4.2	Representação do grafo completo para o IM definido por 3 subpopulações.	38
4.3	Representação de um agente para IM constituídos por 4 ilhas.	40
5.1	Taxa de Convergência Acumulativa - Topologia Aleatória.	53
5.2	Taxa de Convergência Acumulativa - Topologia Roleta.	55
5.3	Taxa de Convergência Acumulativa - Topologia Torneio.	58
5.4	Taxa de Convergência Acumulativa - Topologia Grafo Completo.	60

5.5	Convergência Acumulativa - Topologia Q-learning.	63
5.6	Convergência Acumulativa - Topologia UCB.	66
5.7	Convergência Acumulativa - Topologia K-medoids.	68
5.8	Convergência Acumulativa - Topologias Dinâmicas - 4 Ilhas	69
5.9	Convergência Acumulativa - Topologias Dinâmicas - 8 Ilhas	70
5.10	Convergência Acumulativa - Topologias Dinâmicas - 16 Ilhas	71
5.11	Convergência Acumulativa - Topologias Dinâmicas - 32 Ilhas	72
5.12	Diâmetro Médio dos Grafos.	75
5.13	Média dos Caminhos Mínimos dos Grafos.	76
5.14	Média do Maior Grau de Conexões de Saída.	77

Lista de Tabelas

3.1	Políticas de seleção de indivíduos para o processo de migração.	29
4.1	Medidas de aptidão das possíveis escolhas da <i>Subpopulação 4</i> em um modelo com 4 ilhas.	35
5.1	Parâmetros do algoritmo DE utilizados nos experimentos.	50
5.2	Comparação entre médias (número de funções de avaliação) em relação à topologia Aleatória.	52
5.3	Comparação entre médias (melhores soluções) em relação à topologia Aleatória.	54
5.4	Comparação entre médias (número de funções de avaliação) em relação à topologia Roleta.	54
5.5	Comparação entre médias (melhores soluções) em relação à topologia Roleta.	56
5.6	Parâmetros avaliados para a topologia Torneio.	57
5.7	Comparação entre médias (número de funções de avaliação) em relação à topologia Torneio.	57
5.8	Comparação entre médias (melhores soluções) em relação à topologia Torneio.	57
5.9	Parâmetros avaliados para a topologia Grafo Completo.	59
5.10	Comparação entre médias (número de funções de avaliação) em relação à topologia Grafo Completo.	61

5.11	Comparação entre médias (melhores soluções) em relação à topologia Grafo Completo.	61
5.12	Parâmetros avaliados para a topologia Q-learning.	61
5.13	Comparação entre médias (número de funções de avaliação) em relação à topologia Q-learning.	62
5.14	Comparação entre médias (melhores soluções) em relação à topologia Q-learning.	62
5.15	Parâmetros avaliados para a topologia UCB.	64
5.16	Comparação entre médias (número de funções de avaliação) em relação à topologia UCB.	65
5.17	Comparação entre médias (melhores soluções) em relação à topologia UCB.	65
5.18	Comparação entre médias (número de funções de avaliação) em relação à topologia K-medoids.	67
5.19	Comparação entre médias (melhores soluções) em relação à topologia K-medoids.	67

Nomenclatura

<i>DE</i>	-	Evolução Diferencial
<i>EA</i>	-	Algoritmo Evolutivo
<i>EP</i>	-	Programação Evolutiva
<i>ES</i>	-	Estratégia Evolutiva
<i>GA</i>	-	Algoritmo Genético
<i>GP</i>	-	Programação Genética
<i>GPU</i>	-	Placas de Processamento Gráfico
<i>IM</i>	-	Modelo em Ilhas
<i>NC</i>	-	Computação Natural
<i>MAB</i>	-	Multi-armed Bandit
<i>MAS</i>	-	Sistemas Multiagentes
<i>PSO</i>	-	Otimização por Enxame de Partículas
<i>UCB</i>	-	Upper Confidence Bound

Capítulo 1

Introdução

1.1 Introdução

O aumento da complexidade dos problemas e da capacidade computacional podem ser considerados eventos paralelos e mutuamente estimulantes. A habilidade de manipular grandes quantidades de dados e variáveis, rapidez no cálculo de operações aritméticas e boa capacidade de generalização são características importantes dos atuais computadores. Ao mesmo tempo, tais propriedades são fundamentais para resolução de problemas complexos, não restringindo-se apenas a problemas matemáticos.

A Otimização é um processo que utiliza métodos computacionais para encontrar a melhor solução para um dado problema. Os problemas de otimização aplicados são considerados inerentemente difíceis, pois demandam recursos significativos para sua resolução. Em muitos problemas de otimização, a solução desejada não é conhecida e pertence à um conjunto finito ou infinito de possibilidades. Este processo se inicia através de uma solução primária que melhora gradativamente durante as iterações. Ao final do processo, a melhor solução é apresentada como resposta.

Algoritmos Evolutivos (EA - *Evolutionary Algorithms*) são metaheurísticas que combinam procedimentos para solucionar problemas complexos de otimização (Olafsson 2006). Estas metaheurísticas são projetadas para problemas onde outros métodos de otimização não são eficazes ou eficientes (Olafsson 2006). Desde os anos de 1960, os métodos evolutivos tem sido utilizados para resolução de problemas de otimização (Bäck, Hammel & Schwefel 1997).

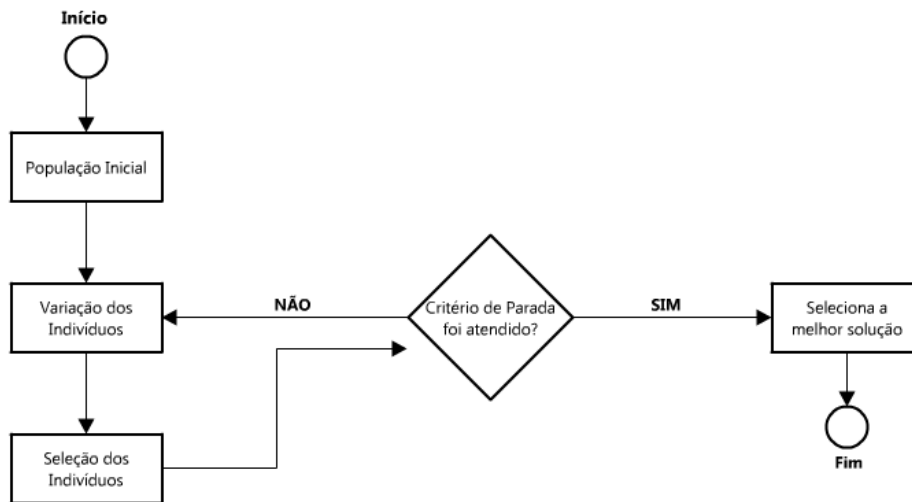


Figura 1.1: Diagrama representativo do fluxo básico dos Algoritmos Evolutivos.

A família dos Algoritmos Evolutivos pertencem a um superconjunto de técnicas denominado Computação Natural (NC - *Natural Computing*) (de Castro 2007). Estas técnicas são baseadas em processos biológicos como seleção natural (Darwin 1872) e herança genética (Schoenauer & Michalewicz 1997). EAs são metaheurísticas populacionais, conhecidas também como métodos de exploração-orientada (Luque & Alba 2011). Sua população inicial de soluções candidatas é aprimorada por meio de variações realizadas em seus indivíduos e pelo processo de seleção, como descreve o esquema básico dos EAs apresentado na Figura 1.1.

Assim como os demais EAs, o método de Evolução Diferencial (DE - *Differential Evolution*) é um algoritmo populacional (Storn & Price 1995), (Storn & Price 1997). Ele foi proposto e relatado por seus autores em meados dos anos 90. Nesta abordagem, novas soluções candidatas são geradas a partir de diferenças vetoriais. Estes vetores diferenciais são formados através de pares de soluções retiradas da própria população de soluções candidatas. No DE, o processo de seleção de uma nova população acontece baseado no valor da função objetivo que representa o problema de otimização.

As razões para o sucesso do DE podem ser explicadas por sua simplicidade e fácil implementação, enquanto demonstra confiabilidade e bom desempenho (Weber, Tirronen & Neri 2010). Além do mais, o método DE se adapta muito bem à paralelização

(Storn & Price 1997). Sua importância pode ser inferida a partir da quantidade de aplicações em que já foi empregado. Tarefas de classificação em mineração de dados (Lopes, Freitas, Silva & Guimarães 2012), agrupamento de dados (Sai, Vinaya, Govardhan & Satapathy 2010), problemas de escalonamento de tarefas (Onwubolu & Davendra 2006) e projeto de filtros digitais (Karaboga 2005) são alguns dos casos de aplicações bem-sucedidas desta metaheurística.

O Modelo em Ilhas (IM - *Island Model*) é uma importante abordagem multipopulacional para melhorar o desempenho dos EAs na resolução de problemas complexos (Cantú-Paz 1998), (Ruciński, Izzo & Biscani 2010). IM são também conhecidos como estratégias de paralelização de granularidade grossa (Belding 1995). Assim, em projetos de IM para Algoritmos Evolutivos, a população total é dividida em subpopulações, conhecidas como *ilhas* (Luque & Alba 2011). Neste modelo, cada ilha executa um EA independentemente e mantém sua própria subpopulação para pesquisa orientada.

No IM, periodicamente ocorre uma troca de indivíduos entre as ilhas. Este processo é conhecido como *migração* e ocorre somente entre ilhas ditas vizinhas (Ruciński, Izzo & Biscani 2010). O processo de migração cria fluxos direcionados entre as ilhas do modelo, denominados *topologias de migração* (Lopes, Silva, Campelo & Guimaraes 2012). Este fluxo migratório forma a estrutura de vizinhança do IM. As topologias de migração representam uma importante regra do IM e podem exercer grande influência no seu desempenho (Ruciński, Izzo & Biscani 2010).

Em geral, topologias de migração podem ser classificadas em dois grandes grupos: *estáticas* e *dinâmicas* (Tang, Lim, Ong & Er 2004), (Berntsson 2006). Em topologias estáticas, o fluxo migratório é definido a priori e mantém-se da mesma forma que foi projetado até o fim do processo (Tang, Lim, Ong & Er 2004). Por outro lado, as topologias dinâmicas podem ser reconfiguradas durante a execução de um IM (Tang, Lim, Ong & Er 2004). Para estas topologias, um mecanismo capaz de controlar e definir o processo de reconfiguração do fluxo migratório do IM se faz necessário.

Encontra-se na literatura alguns trabalhos relacionados às topologias dinâmicas para Modelo em Ilhas (Tang, Lim, Ong & Er 2004), (Berntsson & Tang 2005), (Lardeux & Goëffon 2010), (Lopes, Silva, Campelo & Guimaraes 2012). Em sua maioria, os trabalhos citados anteriormente apresentam mecanismos de configuração da topologia de migração totalmente distintos entre si. Apesar dessas topologias introduzirem características peculiares, elas nunca foram avaliadas em um mesmo estudo.

Neste trabalho, uma análise crítica das topologias de migração dinâmicas para Mo-

delo em Ilhas usando o algoritmo de Evolução Diferencial será apresentado. Além disso, esta análise aponta as principais características dos mecanismos empregados em cada topologia estudada. Seu comportamento e desempenho são analisados e comparados com tradicionais topologias estáticas encontradas na literatura.

1.2 Objetivos

1.2.1 Objetivos Gerais

Estudar topologias de migração dinâmicas para IM usando o algoritmo de Evolução Diferencial. Neste estudo, uma análise crítica será conduzida acerca das características dos mecanismos de configuração usados pelas topologias dinâmicas propostas, assim como, sobre o desempenho delas em um conjunto de problemas de otimização complexos. Espera-se com este trabalho compreender melhor a dinâmica entre os processos evolucionário e migratório em IM baseados no algoritmo de Evolução Diferencial.

1.2.2 Objetivos Específicos

- Estudar topologias de migração dinâmicas propostas na literatura;
- Estudar e caracterizar as principais propriedades dos mecanismos de configuração das topologias de migração dinâmicas apresentadas;
- Estudar os parâmetros de cada topologia de migração dinâmica;
- Avaliar o desempenho das topologias de migração dinâmicas estudadas;
- Avaliar as topologias finais geradas pelas diferentes estratégias;
- Comparar o desempenho de todas as topologias estudadas.

1.3 Motivação

IM é uma interessante e popular estratégia de paralelização para Algoritmos Evolutivos. Seu sucesso pode ser explicado pelo seu bom desempenho e boa capacidade de exploração do espaço de soluções (Ruciński, Izzo & Biscani 2010). Os bons resultados

obtidos pelo IM tornaram esta estratégia uma eficiente forma de lidar com problemas complexos de otimização usando EAs.

A escolha dos parâmetros de configuração do IM são fatores que impactam seu desempenho, como já relataram trabalhos anteriores (Cantú-Paz 1998), (Cantú-Paz & Goldberg 1999), (Cantú-Paz 2001), (Tang, Lim, Ong & Er 2004). Um dos parâmetros críticos para a definição de um IM adequado é a topologia de migração. Ela é responsável por definir o fluxo migratório entre as subpopulações do modelo implementado (Lopes, Silva, Campelo & Guimaraes 2012). A topologia de migração afeta a qualidade da solução final e o ritmo de convergência, seu impacto parece evidente em IM maiores (Ruciński, Izzo & Biscani 2010).

Como mencionado em outros trabalhos, as topologias de migração apresentam efeitos implícitos sobre o IM (Ruciński, Izzo & Biscani 2010), (Lima, Lopes & Tavares 2009), (Miyagi, Tengan, Mohamed & Nakamura 2010). O primeiro deles, um efeito positivo, é a melhora no desempenho causado pela troca de informações entre as ilhas. Outro efeito, algumas vezes um problema, é a sobrecarga de comunicação e processamento causado pelo fluxo de informações requerido.

Apesar da sua importância a maioria das topologias são ingênuas e baseadas principalmente em redes e arquiteturas de computadores (Lopes, Silva, Campelo & Guimaraes 2012), (Cantú-Paz 1998). As topologias estáticas não levam em consideração a dinâmica de exploração implícita e desperdiçam o conhecimento obtido durante o processo de otimização (Lopes, Silva, Campelo & Guimaraes 2012).

Assim, topologias capazes de aprender e adaptar sua estrutura com base em informações sobre o processo evolucionário mostram-se interessantes. Topologias dinâmicas são capazes de utilizar o conhecimento disponível e permitem alterar seu fluxo migratório em tempo real, dependendo do estágio do processo de busca. Estudos já apontam que topologias dinâmicas possibilitam obter melhores resultados que as estáticas (Berntsson & Tang 2005), (Berntsson 2006), (Lardeux & Goëffon 2010).

1.4 Trabalhos Relacionados

Em um de seus estudos, Tang, Lim, Ong & Er (2004) constataram o efeito das topologias de migração sobre o IM, ao avaliar o desempenho de duas topologias distintas. Os autores relataram neste trabalho que a topologia de migração em anel explora melhor

o espaço de busca em comparação à aleatória.

Ruciński, Izzo & Biscani (2010), em um trabalho mais amplo, avaliaram cerca de 14 diferentes topologias de migração, bem como a variação de outros parâmetros do IM e os EAs utilizados. O trabalho menciona a topologia de migração como uma importante regra a ser definida para o uso deste modelo. Segundo Ruciński, Izzo & Biscani (2010), o fluxo migratório entre as ilhas afeta a qualidade da solução final obtida e o ritmo de convergência.

Um dos primeiros trabalhos relatados na literatura, que apresentam uma topologia de migração dinâmica e “inteligente” foi proposto por Berntsson e Tang em 2005 (Berntsson & Tang 2005). Eles definiram um mecanismo responsável por determinar grupos de subpopulações usando o método de agrupamento *K-medoid*. Este mecanismo define grupos de ilhas que estejam trabalhando em uma mesma região do espaço de busca, no momento do processo migratório. Após a determinação dos grupos a topologia de migração é configurada entre ilhas pertencentes ao mesmo grupo ou a grupos distintos.

Alguns anos depois, Lardeux & Goëffon (2010) desenvolveram uma abordagem dinâmica para topologias de migração baseada em recompensa ou penalidade. Nesta estratégia, as possíveis ligações entre as ilhas são representadas por um grafo completo (Lardeux & Goëffon 2010). Cada aresta do grafo possui uma probabilidade de ser escolhida para determinar o fluxo migratório de uma determinada ilha. Suas probabilidades são dinamicamente atualizadas de acordo com o efeito causado por aquele processo de migração.

Em 2012, uma nova estratégia foi proposta por Lopes, Silva, Campelo & Guimaraes (2012), onde o IM é transformado em um Sistema Multiagentes (MAS - Multi-Agent System). O MAS empregado nesta abordagem é capaz de aprender e adaptar as melhores interligações entre as subpopulações. O processo de aprendizado acontece através da experiência obtida sobre processos migratórios anteriores, durante o processo evolucionário. Neste trabalho, o aprendizado é realizado pelo algoritmo de aprendizado por reforço conhecido como *Q-learning*.

1.5 Organização do Trabalho

O restante deste trabalho é organizado da seguinte forma:

O Capítulo 2 apresenta os conceitos básicos sobre Otimização, a origem e as principais

características dos Algoritmos Evolutivos. Este capítulo descreve também o algoritmo de Evolução Diferencial, que será utilizado nos experimentos computacionais propostos. Além do mais, uma descrição detalhada do funcionamento do DE e seus operadores são realizadas em uma seção dedicada ao método.

No Capítulo 3 deste trabalho, descreve-se diferentes estratégias de paralelização para os Algoritmos Evolutivos encontradas na literatura. A maior parte deste capítulo é reservada à discussão de uma estratégia de paralelização específica, denominada Modelo em Ilhas. As propriedades e parâmetros deste modelo, principalmente a topologia de migração, são estudadas.

O Capítulo 4 é destinado a apresentar as abordagens de topologias de migração dinâmicas para Modelo em Ilhas. Diferentes mecanismos que controlam a configuração do fluxo migratório entre as subpopulações do modelo são estudadas. Assim, os seguintes mecanismos baseados em seleção, recompensa e agrupamento são descritos neste capítulo.

No Capítulo 5, apresenta-se um estudo detalhado dos resultados de cada topologia de migração, descritas no capítulo anterior. Estas topologias de migração são avaliadas em termos de qualidade da solução final e número de gerações consumidas. Neste capítulo, as melhores soluções de cada estratégia estudada anteriormente são comparadas com duas tradicionais topologias de migração: anel e estrela. Além do mais, as topologias finais geradas são avaliadas através de medidas de diâmetro, maior grau de saída e caminho médio.

O Capítulo 6 apresenta as conclusões alcançadas por este trabalho. Ao final, as perspectivas consideradas pelo autor são também resumidas.

Capítulo 2

Fundamentos Teóricos

2.1 Otimização

O processo de otimização consiste em encontrar soluções que atendam todas as condições e forneçam valores de mínimo ou máximo de uma função matemática. Neste processo, a solução desejada pertence ao conjunto finito ou infinito de soluções viáveis ao problema. A otimização é um campo muito extenso, onde seus problemas e técnicas de resolução podem ser subdivididas por diferentes aspectos.

Problemas não lineares com variáveis contínuas são um caso particular da otimização. Uma formulação genérica para este tipo de problema, pode ser dada por:

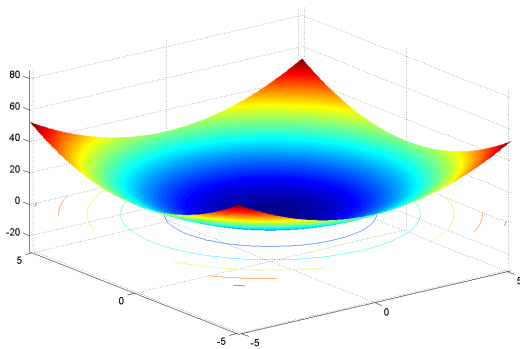
$$\vec{x}^* = \arg \min_{\vec{x}} f(\vec{x}) \quad (2.1)$$

sujeito a:

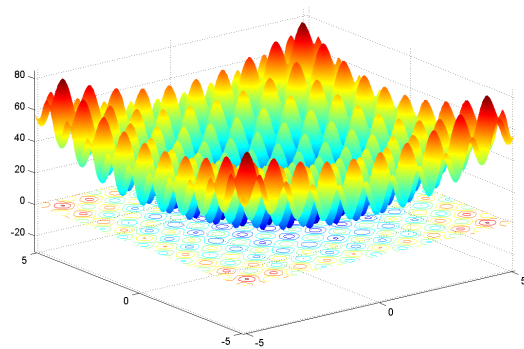
$$\left\{ \begin{array}{l} \vec{x} \in \mathbb{R}^n \\ g_1(\vec{x}) \leq 0 \\ \vdots \\ g_m(\vec{x}) \leq 0 \end{array} \right. \quad (2.2)$$

onde $f(\vec{x})$ representa a medida de custo do problema, conhecida também como *função objetivo*; o vetor \vec{x} corresponde as variáveis de decisão; o vetor \vec{x}^* representa a melhor solução para o problema; as funções de restrição do problema de otimização são representadas por $g_i(\vec{x})$, $i = 1, 2, \dots, m$, onde m é o número máximo de restrições.

Em problemas de otimização não lineares, a função matemática que a descreve, pode ser classificada de acordo com a quantidade de mínimos ou máximos encontrados. Funções unimodais (Rao 2009) apresentam apenas um de pico (máximo) ou vale (mínimo) em um determinado intervalo, como exemplifica a Figura 2.1(a). Assim, as funções classificadas com multimodais apresentam diversos vales ou picos (Rao 2009), vide exemplo da Figura 2.1(b).



(a) Função Sphere - Unimodal



(b) Função Rastrigin - Multimodal

Figura 2.1: Representação gráfica de problemas de otimização bidimensionais.

Em geral, os métodos capazes de resolver problemas de otimização não lineares são dispostos em algoritmos determinísticos, heurísticas e metaheurísticas. A principal característica das técnicas determinísticas é a sequência bem definida de passos. Estas técnicas, em sua maioria, necessitam de pelo menos a primeira derivada da função objetivo, em relação a uma determinada solução. Técnicas determinísticas produzem resultados interessantes apenas em funções contínuas, convexas e unimodais (Bazaraa, Sherali & Shetty 2006). Além disso, a solução encontrada depende da solução inicial.

Heurísticas são técnicas que fornecem boas soluções, sem apresentar garantias formais sobre a qualidade das mesmas. A principal vantagem desta abordagem é a rapidez em que as boas soluções são determinadas. O compromisso entre a eficiência das heurísticas e a precisão da solução final deve ser previamente discutido para este tipo de método.

Metaheurísticas são heurísticas de alto nível que combinam procedimentos, geralmente, outras heurísticas para encontrar boas soluções. Usualmente, as metaheurísticas são algoritmos de propósito geral e não determinísticos (Blum & Roli 2003).

2.2 Computação Natural e Algoritmos Evolutivos

Computação Natural (NC - *Natural Computing*) é uma terminologia criada para representar versões computacionais de processos inspirados da natureza (de Castro 2007). Neste contexto, estes princípios naturais são utilizados como base ao desenvolvimento de novas ferramentas para sistemas computacionais. As novas soluções tecnológicas geradas são empregadas para resolução ou entendimento dos mais diversificados problemas, considerados não-triviais.

Algoritmos Evolutivos (EA - *Evolutionary Algorithm*) são uma classe de metaheurísticas que classificam um subconjunto da NC (de Castro 2007). Os fundamentos da evolução biológica são premissas que orientam as técnicas evolucionárias. O princípio da seleção natural proposto por Charles Darwin, conhecida como Teoria da Evolução (Darwin 1872), é um dos pilares desta classe de algoritmos. Em geral, os EAs são utilizados para desenvolver pesquisas e técnicas de otimização para resolução de problemas complexos (Bäck, Fogel & Michalewicz 2000a), (Bäck, Fogel & Michalewicz 2000b), (Goldberg 1989).

Os EAs são uma categoria de métodos estocásticos que emulam o processo de evolução natural das espécies. Basicamente, EAs mantêm uma coleção de potenciais soluções para o problema, denominados indivíduos. Os indivíduos desta coleção ou população sofrem pequenas variações com o intuito de gerar soluções mais aptas ao problema. A população passa por um processo de pressão seletiva baseado no cálculo da função objetivo. Assim, os indivíduos mais aptos ao problema são aqueles com melhor valor de função objetivo, enquanto soluções ruins apresentam valores piores. O Algoritmo 1 descreve o pseudocódigo básico para os EAs.

Segundo Fogel (1997), os EAs apresentam algumas vantagens sobre os métodos clássicos de otimização numérica. Suas principais características são referenciadas abaixo (Fogel 1997):

1. Simplicidade conceitual;

2. Ampla aplicabilidade;
3. Melhor performance em problemas reais que métodos tradicionais;
4. Robustez a mudanças dinâmicas;
5. Potencial de gerar conhecimento;
6. Hibridização com outros métodos;
7. Facilidade de paralelização;
8. Capacidade de auto-otimização;
9. Capacidade de resolução de problemas para os quais não existem soluções conhecidas.

Algoritmo 1: Algoritmo Evolutivo Básico

```
1 Inicializar população;  
2 Avaliar população;  
3 while not critério de parada do  
4   Selecionar indivíduos;  
5   Gerar nova população;  
6   Avaliar nova população;  
7 Retornar o melhor indivíduo;
```

Apesar das inúmeras vantagens listadas, nos EAs não há garantias que o ótimo global seja determinado em um número finito de iterações (Blickle 1996). A alta demanda computacional e a dificuldade de ajuste dos parâmetros necessários para o bom funcionamento das técnicas evolutivas são outras desvantagens desta abordagem (Blickle 1996). No entanto, provas de convergência assintóticas (Fogel 1994), diferentes formas de paralelização dos EAs (Alba & Tomassini 2002) e a auto-adaptação dos parâmetros (Hinterding, Michalewicz & Eiben 1997) são recursos disponíveis para lidar com seus inconvenientes.

Os métodos de Programação Evolutiva (GP - *Evolutionary Programming*) (Fogel, Owens & Walsh 1966), Estratégia Evolutiva (ES - *Evolution Strategies*) (Rechenberg 1973) e Algoritmo Genético (GA - *Genetic Algorithm*) (Holland 1975) são os pioneiros da classe de EAs. Apesar das resistências encontradas na utilização dessa abordagem, as ideias sobre Computação Natural evoluíram e uma ampla variação dos métodos foram

idealizadas. Dentre as técnicas evolutivas, os GAs e suas variações podem ser considerados um dos mais populares EAs.

Ao longo dos anos, versões aprimoradas dos algoritmos existentes e novas técnicas foram reproduzidas. Na maioria das novas abordagens, os bons aspectos e falhas dos EAs antecessores serviram como inspiração para o desenvolvimento de técnicas mais eficientes e robustas. A Programação Genética (GP - *Genetic Programming*) (Koza 1992), Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) (Kennedy & Eberhart 1995) e Evolução Diferencial (DE - *Differential Evolution*) (Storn & Price 1997) são alguns dos métodos mais atuais encontrados na literatura.

A Evolução Diferencial é considerada um dos atuais estado-da-arte em EAs (Teo 2006), (Das & Suganthan 2011). A Seção 2.3 apresenta o método DE e seus operadores.

2.3 Evolução Diferencial

O algoritmo de Evolução Diferencial é considerado uma técnica de otimização confiável, precisa, robusta e rápida (Salman, Engelbrecht & Omran n.d.). O DE é fácil de implementar (Salman, Engelbrecht & Omran n.d.) e requer poucos parâmetros de ajuste (Paterlini & Krink 2004). Por estas e outras características, este método é visto como um importante e poderoso EA (Silva, Lopes, Freitas & Guimaraes 2012).

Inicialmente, o DE foi proposto em 1995 por Rainer Storn e Kenneth Price, para problemas de otimização global sobre espaços contínuos (Storn & Price 1995), (Storn & Price 1997). Ele tem sido aplicado com sucesso em diversos problemas de otimização mono-objetivo (Mezura-Montes, Velázquez-Reyes & Coello Coello 2006), (Chakraborty 2008) e multiobjetivo (Peng, Dai, Chen & Liu 2007), (Zamuda, Brest, Bošković & Žumer 2007). Além disso, recentemente, ele também foi proposto para resolver problemas de otimização combinatórios (Onwubolu & Davendra 2009), (Prado, Pedrosa Silva, Guimarães & Neto 2010a), (Prado, Pedrosa Silva, Guimarães & Neto 2010b).

O DE é uma técnica evolutiva muito similar a outros EAs, principalmente ao Algoritmo Genético (Salman, Engelbrecht & Omran n.d.). Assim, este algoritmo faz o uso de uma população de soluções candidatas para o problema, no intuito de explorar o espaço de busca e determinar a solução ótima. No DE original, a população de soluções candidatas é gerada aleatoriamente dentro da região de domínio do problema, normalmente descrito como:

$$\mathcal{X} = \{ \vec{x} \in \mathbb{R}^n : x_k^{min} \leq x_k \leq x_k^{max}, k = 1, \dots, D \} \quad (2.3)$$

onde x_k^{min} e x_k^{max} são respectivamente os limites inferiores e superiores de cada variável de decisão e D é a dimensão do problema, ou seja, o número de variáveis do problema de otimização.

Neste trabalho, a seguinte notação $x_{g,i,j}$ será adotada, de tal forma que $g = 1, \dots, G$ representa o contador de gerações; $i = 1, \dots, NP$ representa o índice do indivíduo na população; e $j = 1, \dots, D$ representa o índice da variável de decisão. O valores de G e NP representam o número de máximo de gerações e número de indivíduos na população, respectivamente. Assim, um indivíduo é representado por:

$$\vec{x}_{g,i} = \begin{bmatrix} x_{g,i,1} \\ x_{g,i,2} \\ x_{g,i,3} \\ \vdots \\ x_{g,i,D} \end{bmatrix} \quad (2.4)$$

No DE, novas soluções são geradas através da combinação de indivíduos da própria população de soluções candidatas. A solução construída substitui o indivíduo *atual*, somente se seu valor de função objetivo for melhor que a solução atual. Assim, este algoritmo apresenta três operadores: *mutação diferencial*, *recombinação* e *seleção*.

Embora o DE seja classificado como um EA, e se enquadre em um esquema geral de EAs, seu principal operador não tem base ou inspiração em nenhum processo natural (Guimarães 2009). A maneira como o operador de *mutação diferencial* gera perturbações nos indivíduos na população atual se sustenta em argumentos matemáticos e heurísticos (Guimarães 2009). Contudo, o DE mantém um linha de técnicas evolucionárias, que evoluem uma população de soluções candidatas segundo operadores heurísticos, inspirados

em mecanismos bastante gerais de adaptação natural (Guimarães 2009).

As subseções seguintes descrevem o funcionamento dos operadores de mutação, recombinação e seleção do DE. Além disso, a subseção 2.3.4 apresenta o pseudocódigo do algoritmo de Evolução Diferencial.

2.3.1 Mutação Diferencial

O operador de mutação diferencial é responsável por gerar perturbações nos indivíduos da população atual. As perturbações geradas são baseadas na diferença vetorial entre indivíduos escolhidos aleatoriamente da população atual. Os vetores diferença gerados são multiplicados por uma constante e adicionados a outro indivíduo, conhecido como *vetor base*, que conduz ao *vetor mutante* $\vec{v}_{g,i}$. Quatro estratégias de mutação amplamente utilizadas são descritas abaixo:

- **rand/1**

$$\vec{v}_{g,i} = \vec{x}_{g,r1} + F(\vec{x}_{g,r2} - \vec{x}_{g,r3}) \quad (2.5)$$

- **best/1**

$$\vec{v}_{g,i} = \vec{x}_{g,best} + F(\vec{x}_{g,r1} - \vec{x}_{g,r2}) \quad (2.6)$$

- **current-to-best/1**

$$\vec{v}_{g,i} = \vec{x}_{g,i} + F(\vec{x}_{g,best} - \vec{x}_{g,i}) + F(\vec{x}_{g,r1} - \vec{x}_{g,r2}) \quad (2.7)$$

- **current-to-rand/1**

$$\vec{v}_{g,i} = \vec{x}_{g,i} + F(\vec{x}_{g,r1} - \vec{x}_{g,i}) + F(\vec{x}_{g,r2} - \vec{x}_{g,r3}) \quad (2.8)$$

onde $r_1 \neq r_2 \neq r_3 \neq i \in \{1, 2, \dots, NP\}$ são índices aleatórios mutuamente distintos; *best* é o índice da melhor solução na população atual; e F é um fator escalar aplicado ao vetor diferença. O parâmetro F é responsável por controlar a amplificação entre os vetores.

Para cada $\vec{x}_{g,i}$ na população, uma solução mutante $\vec{v}_{g,i}$ é gerada. Assim, a Figura 2.2 exemplifica o procedimento do operador de mutação diferencial. Neste exemplo, a

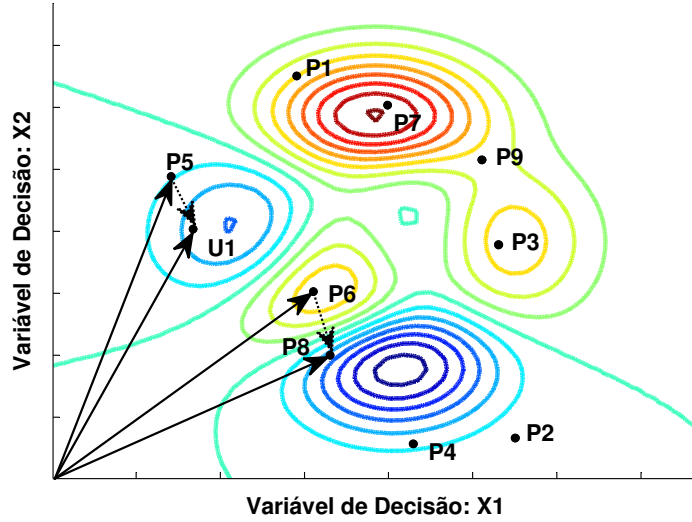


Figura 2.2: Ilustração do procedimento de geração de uma solução mutante.

solução $P5$ é escolhida como vetor base e os vetores de solução $P6$ e $P8$ são utilizados na diferença vetorial. Após aplicação da Equação 2.5, a solução \vec{u}_1 é produzida.

2.3.2 Recombinação

No operador de recombinação, um vetor teste $\vec{u}_{g,i}$ é produzido através da recombinação do vetor atual $\vec{x}_{g,i}$ e o vetor mutante $\vec{v}_{g,i}$. Na versão clássica do algoritmo DE, uma recombinação discreta com probabilidade CR é utilizada, como podemos ver no seguinte esquema:

$$u_{g,i,j} = \begin{cases} v_{g,i,j}, & \text{se } \mathcal{U}_{[0,1]} \leq CR \text{ or } j = k \\ x_{g,i,j}, & \text{caso contrário} \end{cases} \quad (2.9)$$

onde $\mathcal{U}_{[0,1]}$ representa a amostragem de uma variável aleatória com distribuição uniforme entre $[a, b]$ e k representa um índice escolhido aleatoriamente entre $[1, D]$, lembrando que D representa o número de variáveis de decisão do problema.

As Equações 2.10 e 2.11 representam, respectivamente, soluções genéricas de um vetor atual e um vetor mutante, considerando um problema de otimização com 6 variáveis de

decisão. A Equação 2.12, ilustra uma possível solução teste gerada pelo procedimento de recombinação do DE.

$$\vec{x}_{g,i} = \langle x_{g,i,1}, x_{g,i,2}, x_{g,i,3}, x_{g,i,4}, x_{g,i,5}, x_{g,i,6} \rangle \quad (2.10)$$

$$\vec{v}_{g,i} = \langle v_{g,i,1}, v_{g,i,2}, v_{g,i,3}, v_{g,i,4}, v_{g,i,5}, v_{g,i,6} \rangle \quad (2.11)$$

$$\vec{u}_{g,i} = \langle v_{g,i,1}, x_{g,i,2}, v_{g,i,3}, v_{g,i,4}, x_{g,i,5}, v_{g,i,6} \rangle \quad (2.12)$$

2.3.3 Seleção

O operador de seleção é responsável por determinar os indivíduos mantidos na população seguinte. Neste operador, a solução teste $\vec{u}_{g,i}$ compete com a solução atual $\vec{x}_{g,i}$, baseado em seus valores de função objetivo. Se o vetor teste é melhor ou igual ao vetor atual, ele substitui a solução atual. Caso contrário, a solução atual sobrevive, enquanto a solução teste é eliminada, como descreve o esquema abaixo:

$$\vec{x}_{g+1,i} = \begin{cases} \vec{u}_{g,i}, & \text{se } f(\vec{u}_{g,i}) \leq f(\vec{x}_{g,i}) \\ \vec{x}_{g,i}, & \text{caso contrário} \end{cases} \quad (2.13)$$

Variações deste esquema apresentado anteriormente são discutidas em: (Price, Storn & Lampinen 2005), (Mezura-Montes, Velázquez-Reyes & Coello Coello 2006), (Chakraborty 2008).

2.3.4 Pseudocódigo

O Algoritmo 2 apresentado abaixo, descreve os passos da versão clássica do algoritmo de Evolução Diferencial. Mais detalhes podem ser encontrados em: (Storn & Price 1995), (Storn & Price 1997).

Algoritmo 2: Algoritmo de Evolução Diferencial

```

1  $g \leftarrow 1$  ;
2 Inicializar população  $X_g = \{x_{g,i}, i = 1, \dots, NP\}$ ;
3 Avaliar população ( $X_g$ );
4 while not critério de parada do
5   for  $i = 1$  to  $NP$  do
6     Escolher aleatoriamente  $r_1, r_2, r_3 \in 1, \dots, NP$ ;
7     Escolher aleatoriamente  $k \in 1, \dots, D$ ;
8     for  $j = 1$  to  $D$  do
9       if  $\mathcal{U}_{[0,1]} \leq CR$  or  $j = k$  then
10        |  $u_{g,i,j} = x_{g,r1,j} + F^*(x_{g,r2,j} - x_{g,r3,j})$ ;
11        else
12        |  $u_{g,i,j} = x_{g,i,j}$  ;
13   for  $i = 1$  to  $NP$  do
14     if  $f(u_{g,i}) \leq f(x_{g,i})$  then
15       |  $x_{g+1,i} \leftarrow u_{g,i}$ ;
16     else
17       |  $x_{g+1,i} \leftarrow x_{g,i}$ ;
18    $g \leftarrow g + 1$  ;
19 Retornar o melhor indivíduo;

```

Capítulo 3

Estratégias de Paralelização e o Modelo em Ilhas

3.1 Estratégias de Paralelização

Em geral, EAs são capazes de encontrar boas soluções em tempos computacionais aceitáveis. No entanto, o aumento da dimensionalidade e da complexidade dos problemas de otimização elevam o tempo de processamento gasto por estes métodos populacionais (Cantú-Paz 1998). Por mais interessante que seja a solução encontrada pelos EAs, tempos exorbitantes não são aceitáveis devido ao processo decisório que evolui os problemas de otimização.

Estratégias de paralelização são promissoras formas de manter metaheurísticas populacionais competitivas diante de problemas complexos (Adamidis 1998), (Cantú-Paz 1998). Segundo Tomassini (1999), há duas razões principais para a paralelização de um EA. A primeira delas é alcançar uma economia de tempo distribuindo o esforço computacional. A segunda razão é beneficiar-se de uma configuração paralela a partir do ponto de vista algorítmico, ou seja, da evolução natural paralela de populações espacialmente distribuídas.

Por padrão os EAs são inerentemente paralelos e distribuídos (Alba & Tomassini 2002). Uma das primeiras investigações sobre modelos paralelos de EAs foram realizadas por Bethke (1976). Neste trabalho, ele descreveu a implementação de um modelo paralelo para GAs convencionais com troca parcial de indivíduos em sucessivas gerações. Seus resultados indicaram grande eficiência dos GAs paralelos, mas o autor ignora em

sua análise final os tempos de comunicação do modelo (Cantú-Paz n.d.).

Atualmente, dispositivos computacionais de alto desempenho se tornaram realidade e de fácil acessibilidade. Este acontecimento garantiu a aplicabilidade das estratégias de paralelização para os EAs. O sucesso e o reconhecimento dos EAs paralelos podem ser comprovados pelos inúmeros trabalhos encontrados na literatura (Alba 2002), (Tasoulis, Pavlidis, Plagianakos & Vrahatis 2004). Tem-se relato da utilização de multiprocessadores (Golub & Budin 2000), placas gráficas (GPU - *Graphics Processing Unit*) (Luong, Melab & Talbi 2010) e até mesmo supercomputadores na implementação de EAs paralelos (Gulsen & Smith 1999).

Os EAs paralelos podem ser facilmente projetados por três diferentes estratégias de paralelização. A mais intuitiva dessas estratégias é projetada para paralelização da função de avaliação, ou seja, do cálculo da medida de custo dos EAs (Cantú-Paz 1998), (Alba & Troya 1999). O processo de avaliação dos indivíduos é a tarefa mais custosa nos EAs (Storn & Price 1997). Assim, neste projeto a tarefa de avaliar toda a população de soluções é dividida entre todos os processos (unidades de processamento ou computadores) disponíveis.

A Figura 3.1 exemplifica graficamente o projeto de paralelização do cálculo da medida de custo para um EA. Este projeto ilustrado considera uma região de memória compartilhada entre as unidades de processamento disponíveis. Esta região de memória é utilizada para disponibilizar os vetores de soluções e os resultados do cálculo da avaliação da função de custo pelos processos. O primeiro passo é definir as tarefas que serão realizadas por cada unidade de processamento. Após o processamento das tarefas os processos são todos sincronizados.

Baixa dependência, sincronização e comunicação entre as tarefas são características que garantem o sucesso desta estratégia. No entanto, o *SpeedUp*¹ máximo alcançado por esta estratégia é linear, isto porque nenhuma alteração no funcionamento do EA é realizada. *SpeedUp* linear é atingido quando o tempo gasto pelo método em paralelo é igual ao tempo gasto pelo método sequencial dividido pelo número de unidades de processamento utilizadas.

Esta abordagem de paralelização de EAs, descrita anteriormente, não se limita apenas a divisão do esforço computacional realizado para calcular a medida de custo. É possível encontrar na literatura diversos trabalhos que aplicaram esta estratégia em ou-

¹*SpeedUp*: Medida de desempenho de métodos paralelos (Dongarra, Foster, Fox, Gropp, Kennedy, Torczon & White 2003).

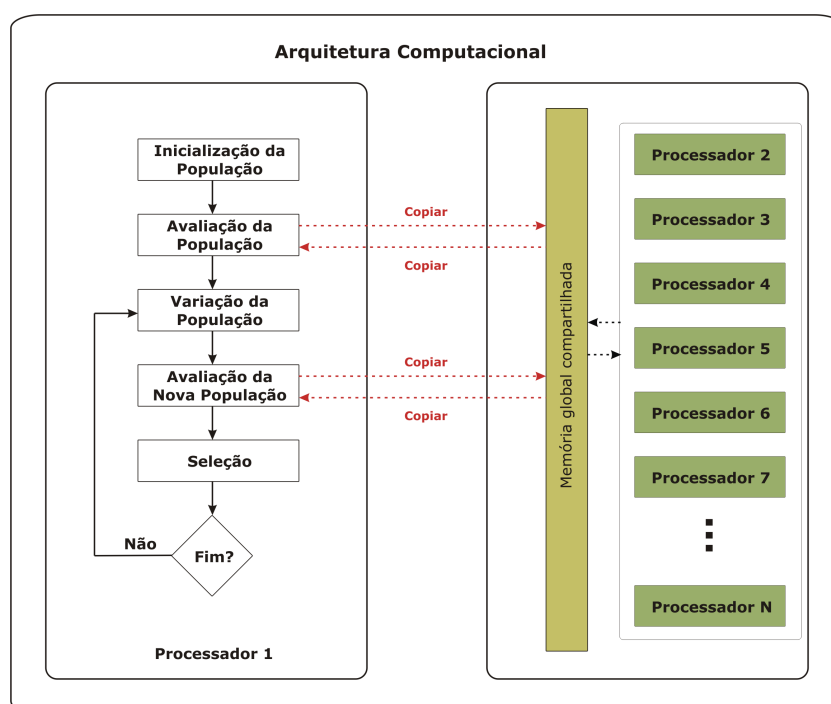


Figura 3.1: Projeto de paralelização do método de avaliação das soluções de um EA.

tros operadores dos EAs, vide (Luong, Melab & Talbi 2010), (Veronese & Krohling 2010). De uma forma geral, estes projetos utilizam uma estrutura semelhante à apresentada na Figura 3.1.

As outras estratégias de paralelização de EAs existentes são denominadas Modelo em Ilhas e Modelo Híbrido. A abordagem de Modelo em Ilhas será apresentada na Seção 3.2. O Modelo Híbrido é uma combinação das abordagens apresentadas e não será discutida neste trabalho. Detalhes sobre o funcionamento deste modelo podem ser encontrados nos seguintes trabalhos (Gorges-Schleuter 1989), (Gorges-Schleuter 1990), (Loukil, Mehdi, Melab, Talbi & Bouvry 2009).

3.2 Modelo em Ilhas

O Modelo em Ilhas é uma importante abordagem multipopulacional para EAs (Cantú-Paz 1998). Esta abordagem tornou-se uma maneira popular e eficiente de implementar EAs em arquiteturas computacionais paralelas ou sequenciais (Rudolph 1991), (Cantú-Paz 1998), (Whitley, Rana & Heckendorn 1998), (Tasoulis, Pavlidis, Plagianakos &

Vrahatis 2004). Sua inspiração é fundamentada originalmente da teoria do Equilíbrio Pontuado (Cohoon, Hegde, Martin & Richards 1987). Esta teoria foi proposta com o intuito de explicar certos dilemas paleontológicos ao longo do registro geológico.

Um dos primeiros estudos sobre IM foi realizado por Pettey, Leuze & Grefenstette (1987). Inicialmente, este modelo foi proposto para paralelização de Algoritmos Genéticos (Pettey, Leuze & Grefenstette 1987), (Cohoon, Hegde, Martin & Richards 1987). No entanto, o IM tem sido aplicado com sucesso em outras metaheurísticas populacionais como o algoritmo de Evolução Diferencial (Tasoulis, Pavlidis, Plagianakos & Vrahatis 2004) e Estratégias Evolutivas (Rudolph 1991).

Basicamente, a ideia do Modelo em Ilhas é dividir a população de soluções candidatas em populações menores, veja o exemplo da Figura 3.2. Neste modelo, estas subpopulações permanecem relativamente isoladas e são conhecidas como *ilhas*. Cada ilha definida utiliza um EA para evoluir independentemente sua própria população de soluções. Diferentes EAs podem ser projetados para cada ilha do modelo.

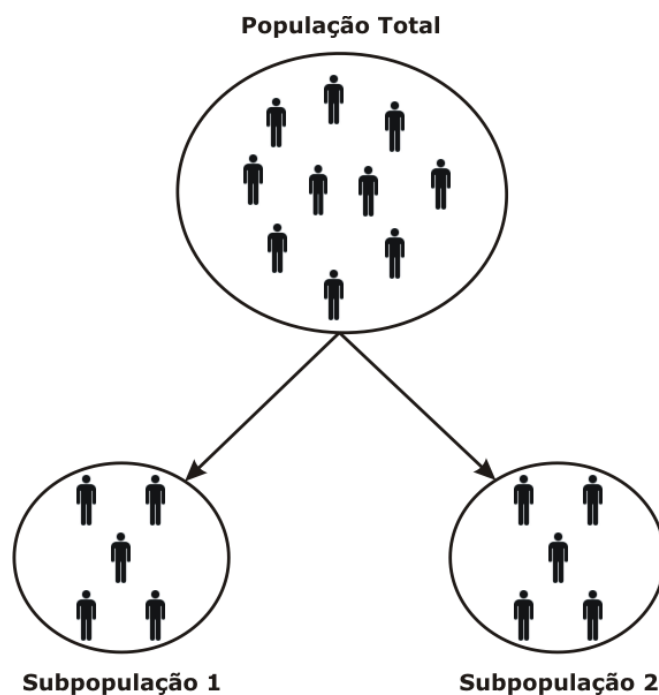


Figura 3.2: Processo de divisão da população total em subpopulações.

O relativo isolamento entre as ilhas promove uma “saudável” competição entre elas. Através desta concorrência é possível que ilhas distintas realizem buscas em regiões diferentes do espaço de soluções, vide Figuras 3.3(a) e 3.3(b). Embora a competição

seja um aspecto positivo, a colaboração é fundamental para se alcançar o objetivo final. Para promover esta colaboração entre as ilhas, o *processo de migração* acontece periodicamente na execução do IM.

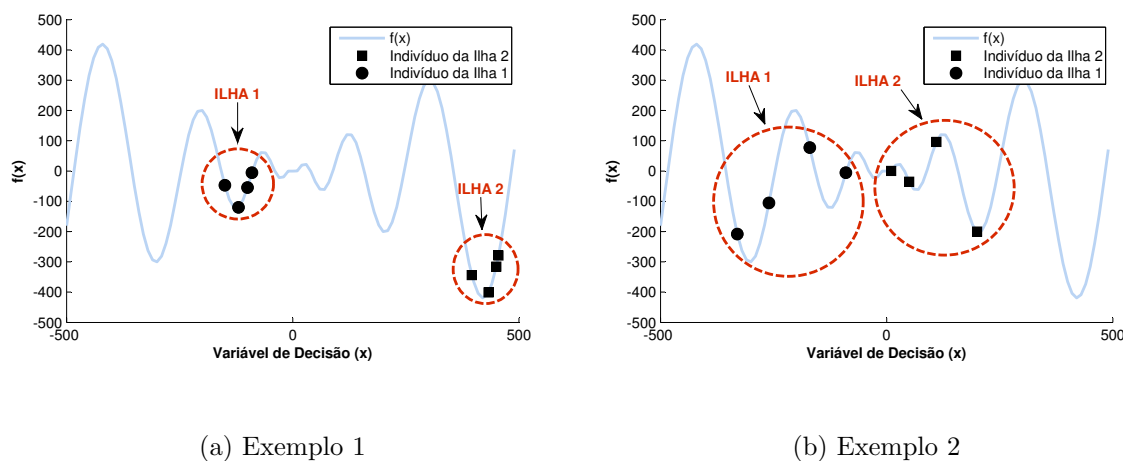


Figura 3.3: Ilustração da distribuição espacial das subpopulações para o problema de otimização Schwefel unidimensional.

No processo de migração, a cada período pré-determinado uma troca de informação entre as ilhas é realizada. A migração permite que subpopulações com maior aptidão colaborem na evolução das ilhas menos desenvolvidas (Lopes, Silva, Campelo & Guimaraes 2012). Este efeito é possível por meio das novas informações sobre o problema de otimização inseridas nestas subpopulações.

A Figura 3.4 ilustra o funcionamento básico do processo de migração. Neste procedimento, *indivíduos migrantes* são selecionados em cada subpopulação. Estes indivíduos são copiados e enviados para as subpopulações nas quais foram destinados. Cada ilha que recebe um indivíduo migrante seleciona uma solução de sua população para ser substituída pelo indivíduo recebido. Ao término da substituição dos indivíduos em cada subpopulação o processo de migração é finalizado.

Devido às suas características, o IM é facilmente paralelizável (Cantú-Paz 2001), (Lopes, Silva, Campelo & Guimaraes 2012). Cada EA responsável por evoluir uma determinada subpopulação pode ser executado em uma unidade de processamento. A comunicação entre estas unidades de processamento é somente requisitada durante o processo de migração. Assim, IM também apresenta baixa comunicação e sincronização.

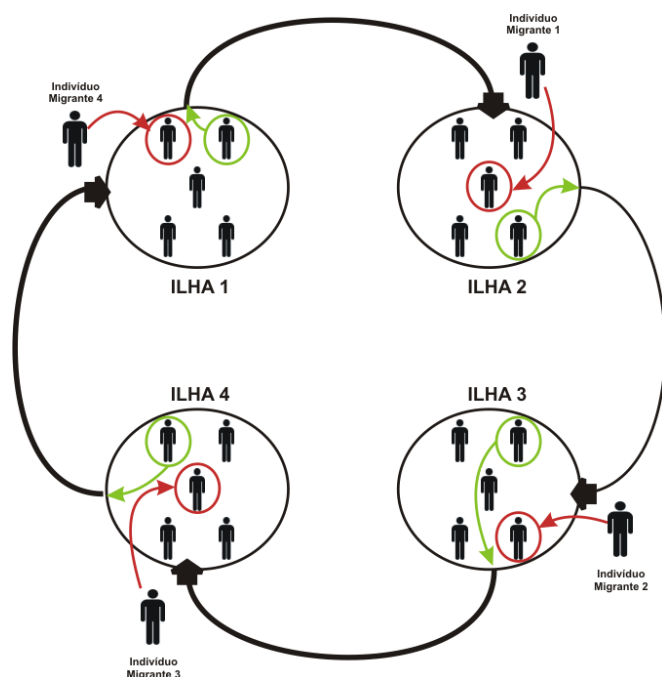


Figura 3.4: Processo migratório para IM com 4 topologias usando topologia em anel.

A principal vantagem do IM é a melhoria da busca no espaço de soluções (Ruciński, Izzo & Biscani 2010). Sua capacidade de balancear o processo de busca entre exploração do espaço e refinamento da solução é benéfico ao processo evolutivo em si (Lopes, Silva, Campelo & Guimaraes 2012). Assim, mesmo em versões sequenciais, o IM melhora a qualidade da solução e o tempo de convergência dos EAs (Tasoulis, Pavlidis, Plagianakos & Vrahatis 2004). Esta melhora possibilita ganhos de desempenho em relação a versão clássica dos EAs e alguns trabalhos afirmam que sua execução pode atingir *SpeedUp* superlinear² (Cantú-Paz 2001), (Alba 2002).

Para um bom comportamento do IM, o desenvolvedor deve definir alguns parâmetros. Estes parâmetros estão diretamente relacionados ao desempenho do modelo. As subseções seguintes apresentam cada um destes parâmetros do IM.

²*SpeedUp* superlinear: significa que o uso de “p” processadores conduz a um método mais rápido do que “p” vezes que sua versão sequencial.

3.2.1 Número de Ilhas

O parâmetro número de ilhas é um dos primeiros a ser definido pelo desenvolvedor. Ele define o número de ilhas ou subpopulações presentes no modelo. Geralmente, seu valor está intrinsecamente ligado à quantidade de unidades de processamento disponíveis para sua execução. Assim, em grande parte dos trabalhos encontrados na literatura, o número de ilhas foi definido por 2^p , sendo $p = \{p \in \mathbb{I} \mid p \geq 1\}$ (Ruciński, Izzo & Biscani 2010), (Luque & Alba 2011), (Lopes, Silva, Campelo & Guimaraes 2012).

Em IM, o número de ilhas possui relação direta com o tamanho da população em cada ilha do modelo. Desta forma, o tamanho de cada subpopulação é dado pela seguinte equação:

$$\text{NP_ILHAS} = \frac{\text{NP_TOTAL}}{\text{NUM_ILHAS}} \quad (3.1)$$

onde NP_TOTAL é a quantidade total de indivíduos do IM e NUM_ILHAS é o número de ilhas definido para o modelo.

Outra forma de configuração é determinar um número fixo de indivíduos por ilha do modelo. Desta maneira, a quantidade total de indivíduos do IM é descrita por:

$$\text{NP_TOTAL} = \text{NUM_ILHAS} \times \text{NP_ILHAS} \quad (3.2)$$

Subpopulações que apresentam quantidades de indivíduos diferentes são ditas *ilhas desbalanceadas*. Esta característica é indesejável em IM implementado em arquiteturas paralelas. A falta de balanceamento pode causar perda de desempenho computacional do IM, pois determinadas ilhas gastarão mais tempo por ciclo (geração) do que outras.

3.2.2 Frequência de Migração

Este parâmetro é responsável por definir a frequência de realização do processo de migração durante a execução do IM projetado. Existem duas formas possíveis de configurar esta frequência de migração. Ela pode ser definida usando um intervalo ou uma

probabilidade de migração (Luque & Alba 2011).

Intervalos de migração descrevem a periodicidade do processo de troca de informações entre as ilhas do modelo (Luque & Alba 2011). Por exemplo, intervalos de migração iguais à 10 indicam que os EAs irão evoluir suas populações durante um ciclo de 10 gerações. Após este ciclo, o processo migratório acontece e um novo ciclo se inicia até que o critério de parada do modelo seja atendido.

A probabilidade de migração decide a cada geração se o processo de migração ocorrerá ou não (Luque & Alba 2011). Neste processo, uma variável $P_m = \{P_m \in \mathbb{R} \mid 0 \leq P_m \leq 1\}$ que representa a probabilidade de ocorrer o processo migratório deve ser definida. Assim, a cada geração uma variável gerada aleatoriamente com distribuição uniforme no intervalo $[0, 1]$ é comparada com a probabilidade de migração P_m . Caso P_m seja maior ou igual à variável gerada, o processo de migração acontece, caso contrário o processo não acontece. Estes passos se seguem até que o critério de parada do modelo seja atendido.

3.2.3 Taxa de Migração

O parâmetro taxa de migração define o número de *indivíduos migrantes*. Estes indivíduos são selecionados para migrarem de uma subpopulação para outra. Alta taxa de migração e frequência pode levar o IM a perder suas boas propriedades e não mais comportar-se da forma esperada (Tanese 1989).

3.2.4 Tipo de Sincronização

Tipo de sincronização define o gerenciamento utilizado para a realização do processo de migração em arquiteturas computacionais paralelas. Este procedimento de sincronização entre as ilhas do modelo pode ser executado de maneira assíncrona ou síncrona (Ruciński, Izzo & Biscani 2010), (Luque & Alba 2011).

Em processos de migração síncronos, a troca de indivíduos ocorre ao mesmo tempo para todas as ilhas do IM (Ruciński, Izzo & Biscani 2010). Além disso, durante este processo migratório nenhum cálculo é realizado. Modelos síncronos são mais fáceis de implementar, entretanto, são mais custosos computacionalmente.

Para processos de migração assíncronos, as subpopulações delegam os indivíduos migrantes tão logo estejam prontos para fazê-los (Ruciński, Izzo & Biscani 2010). Este

procedimento não leva em consideração o estado das demais ilhas. Diferentemente do modelo síncrono, esta estratégia é mais complexa mas pode alcançar melhor desempenho.

3.2.5 Política de Migração

Política de migração define a estratégia empregada para selecionar os indivíduos migrantes. Este parâmetro também é responsável por determinar as soluções que serão substituídas durante o processo de migração. As soluções substituídas são denominadas neste trabalho como *indivíduo substituído*. Existem diversas políticas de migração descritas na literatura e as mais conhecidas são apresentadas na tabela abaixo:

Política de Migração	Indivíduo Migrante	Indivíduo Substituído
Melhor/Aleatório	Melhor solução	Solução aleatória
Melhor/Pior	Melhor solução	Pior solução
Aleatório/Aleatório	Solução aleatória	Solução aleatória
Aleatório/Pior	Solução aleatória	Pior solução

Tabela 3.1: Políticas de seleção de indivíduos para o processo de migração.

Assim como outros parâmetros do IM, a política de migração exerce grande influência no desempenho destes modelos. Segundo Cantú-Paz (2001), a escolha de indivíduos por aptidão no processo migratório aumenta a pressão seletiva no IM. O aumento da pressão seletiva pode levar o IM convergir significativamente mais rápido (Cantú-Paz 2001).

3.2.6 Topologia de Migração

Topologia de migração define o fluxo migratório entre as subpopulações de um IM (Lopes, Silva, Campelo & Guimaraes 2012). Este fluxo é determinado por um conjunto de arestas direcionadas que formam a estrutura de vizinhança do modelo. As arestas desta estrutura representam conexões que interligam duas ilhas que se interagem. Por meio destas conexões as ilhas podem trocar soluções entre si, no processo denominado anteriormente como migração.

Este parâmetro cumpre um importante papel no desempenho do IM (Ruciński, Izzo & Biscani 2010) (Lopes, Silva, Campelo & Guimaraes 2012). A utilização de topologias

de migração adequadas podem garantir modelos com melhores desempenhos. Todavia, a definição da estrutura ou funcionamento destas topologias ainda representa um problema em aberto.

Existem diversas topologias de migração descritas na literatura (Tang, Lim, Ong & Er 2004), (Berntsson & Tang 2005), (Ruciński, Izzo & Biscani 2010), (Lardeux & Goëffon 2010). De uma forma geral, as topologias de migração podem ser divididas em estáticas e dinâmicas. Esta pequena taxonomia distingue topologias onde o fluxo migratório é considerado inalterável ou alterável durante o processo evolutivo.

Topologias de migração estáticas tem seus fluxos definidos a priori e não se alteram durante o processo de evolutivo (Tang, Lim, Ong & Er 2004). Elas são baseadas principalmente em redes e arquiteturas de computadores. Estas topologias são ingênuas e podem sofrer perda de desempenho em IM maiores, pois boas soluções podem levar vários ciclos para serem amplamente propagadas.

A Figura 3.5 ilustra duas tradicionais topologias de migração estáticas para modelos que com quatro ilhas. A mais conhecida topologia de migração é apresentada na Figura 3.5(a). Devido ao seu formato de um círculo fechado, ela foi denominada topologia em anel. A outra topologia descrita, conhecida como topologia em estrela, configura-se por uma ilha central que se conecta com as demais, veja a Figura 3.5(b).

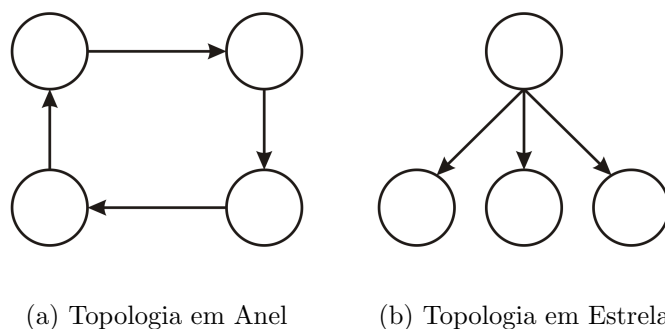


Figura 3.5: Tradicionais topologias de migração estáticas.

As topologias de migração dinâmicas, foco de estudo deste trabalho, são apresentadas no Capítulo 4.

Capítulo 4

Topologias Dinâmicas para Modelo em Ilhas

4.1 Características

Topologias de migração dinâmicas são capazes de modificar seu fluxo migratório a cada estágio do processo evolucionário (Lopes, Silva, Campelo & Guimaraes 2012). Elas são compostas por um mecanismo “inteligente” responsável por fazer emergir uma topologia momentânea ideal. A topologia definida é fruto da crença de seu próprio mecanismo projetado e, ao longo deste trabalho, este mecanismo será tratado como *mecanismo de configuração*.

Os mecanismos de configuração são heurísticas que definem o fluxo migratório entre as ilhas do modelo. Estas heurísticas adotam um ou mais indicadores previamente determinados. Os indicadores são utilizados pelas heurísticas como guia na configuração da melhor topologia momentânea. Em geral, estes indicadores exploram as duas principais características do IM: diversidade populacional e pressão seletiva.

As topologias de migração dinâmicas tem como objetivo melhorar o desempenho e a flexibilidade do IM (Lopes, Silva, Campelo & Guimaraes 2012). Estas topologias são capazes de assimilar o conhecimento gerado durante o processo evolutivo do modelo. O conhecimento adquirido pode ser traduzido em fluxos migratórios momentâneos mais eficientes que as topologias tradicionais. Além do mais, as topologias definidas são independentes do problema de otimização e das configurações do IM.

Este tipo de estratégia de configuração do fluxo migratório requer a existência de meios físicos de comunicação entre todas as ilhas do modelo. Elas também adicionam processamento ao IM devido à execução do mecanismo de configuração. Entretanto, estas desvantagens podem ser compensadas por uma exploração do espaço de busca mais eficiente.

O restante deste capítulo descreve as topologias de migração dinâmicas estudadas. Para uma melhor apresentação, as topologias dinâmicas foram divididas de acordo com as principais características de seus mecanismos de configuração. Desta maneira, as topologias dinâmicas são separadas por topologias baseadas em métodos de seleção, métodos de recompensa e técnicas de agrupamento.

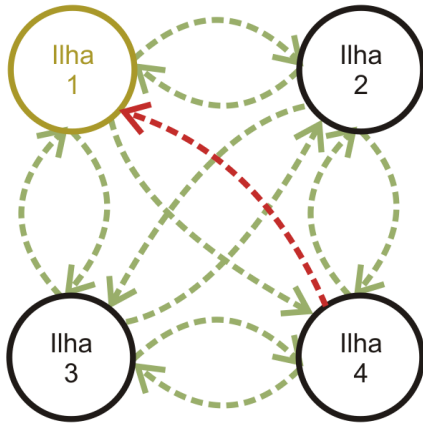
4.2 Topologias baseadas em Métodos de Seleção

4.2.1 Seleção Aleatória

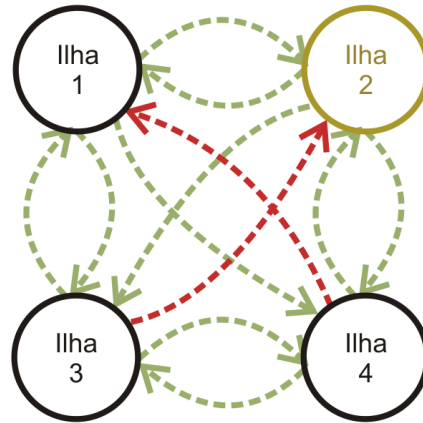
A mais tradicional topologia de migração dinâmica é conhecida como *topologia aleatória* (Tang, Lim, Ong & Er 2004). Seu mecanismo de configuração é totalmente dinâmico e estocástico. O IM é representado por esta abordagem como um grafo completo. Os nodos deste grafo simbolizam as ilhas do modelo. As arestas orientadas representam possíveis migrações que descrevem a origem e o destino dos indivíduos migrantes.

O processo de configuração da topologia de migração é realizado por ilha do modelo. Para cada ilha, a origem da solução migrante recebida é determinada aleatoriamente pelo mecanismo de configuração. Neste processo, os indivíduos migrantes não podem ser enviados para sua própria subpopulação de origem. Assim, ao final do processo de seleção, o fluxo migratório completo do IM é estabelecido.

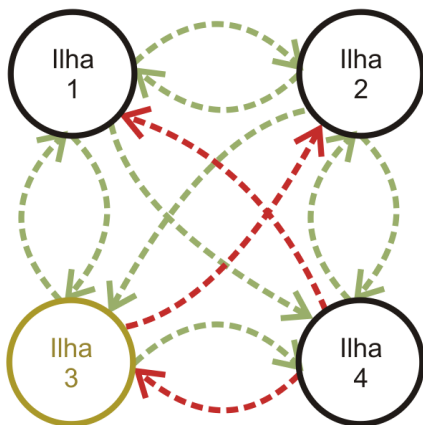
A Figura 4.1 apresenta um diagrama do funcionamento do mecanismo das topologias aleatórias. Em IM com 4 subpopulações são necessários 4 passos para definição total da topologia de migração. Nas figuras descritas neste esquema, as arestas representam possíveis fluxos entre as ilhas. As arestas vermelhas descrevem os fluxos definidos pelo mecanismo de configuração. Desta maneira, a Figura 4.1(d) demonstra a origem de todas soluções recebidas por cada subpopulação do IM.



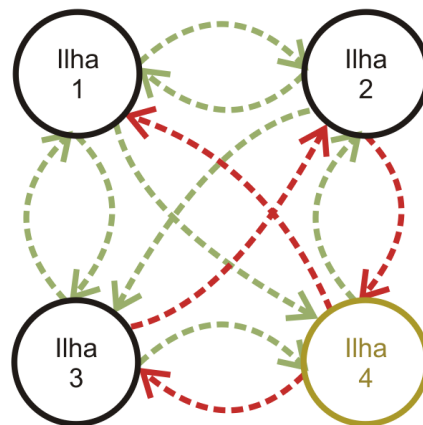
(a) Configuração da Topologia Aleatória: Passo 1



(b) Configuração da Topologia Aleatória: Passo 2



(c) Configuração da Topologia Aleatória: Passo 3



(d) Configuração da Topologia Aleatória: Passo 4

Figura 4.1: Processo de definição da topologia de migração aleatória em um modelo composto por 4 ilhas.

Topologias de migração aleatórias se destacam pela simplicidade do seu mecanismo de configuração. Sua constante alteração do fluxo migratório garante uma boa dinâmica na iteração entre as ilhas do modelo. Além do mais, ela é facilmente implementada e apresenta baixo custo computacional em sua execução.

O mecanismo empregado nesta topologia de migração é considerado muito ingênuo. Ele não toma em consideração informações sobre a qualidade dos indivíduos migrantes e a evolução das subpopulações do IM (Lopes, Silva, Campelo & Guimaraes 2012). O histórico das escolhas passadas também não é mantido por este mecanismo.

4.2.2 Seleção por Roleta

Topologias de migração empregando mecanismos de seleção por roleta são versões mais perspicazes das topologias aleatórias. Seu mecanismo de configuração passa a contar com um método de seleção probabilístico, denominado *seleção por roleta*. Este método de seleção determina os indivíduos selecionados baseados em sua aptidão (Jaddan, Rajamani & Rao 2005).

Esta abordagem funciona de forma semelhante à topologia aleatória. Elas representam o IM da mesma maneira, apenas os mecanismos de configuração são diferenciados. Este mecanismo considera que indivíduos mais aptos tem maior probabilidade de melhorar o desempenho do IM. Assim, para cada ilha a roleta é “girada” e um indivíduo migrante é selecionado. Todas as soluções migrantes participam da seleção, exceto os indivíduos da própria ilha que receberá as soluções.

O processo de escolha do fluxo de migração requer o cálculo da porcentagem de seleção das soluções participantes. Para cada solução participante x_i , esta porcentagem é calculada pela seguinte fórmula:

$$ps = \frac{f(x_i)}{\sum_{j=1}^{ni} f(x_j)} \quad (4.1)$$

onde $f(x_i)$ representa o valor da função de custo para a solução x_i e ni o número total de indivíduos participantes do processo de seleção por roleta.

Logo após ao cálculo das porcentagens, um intervalo de seleção é criado utilizando os valores calculados anteriormente. Este intervalo define as faixas de valores em que as soluções migrantes são escolhidas, proporcionalmente à sua taxa de seleção. Assim, o

intervalo completo incluindo todas as soluções é contido entre 0 e 1.

Este mecanismo de configuração escolhe um valor aleatório no intervalo $[0, 1]$ e verifica em qual faixa o valor atribuído se encontra. O indivíduo que se refere àquela faixa é a solução escolhida pelo método. Maiores detalhes sobre a implementação do método de seleção por roleta são apresentados no Pseudocódigo 3.

Algoritmo 3: Algoritmo de Seleção por Roleta

```

1  $ni \leftarrow$  número de indivíduos;
2  $total \leftarrow \sum_{i=1}^{ni} f(x_i)$ ;
3  $rand \leftarrow$  valor aleatório entre  $[0, 1]$ ;
4  $s \leftarrow 0.0$ ;
5 for  $i = 1$  to  $NI$  do
6    $ps \leftarrow f(x_i)/total$ ;
7    $s \leftarrow s + ps$ ;
8   if  $s \geq rand$  then
9     Escolhe indivíduo atual  $i$ ;
10     $i \leftarrow ni + 1$ ;
11 Retornar o indivíduo selecionado;
```

A Tabela 4.1 apresenta uma simulação hipotética desta seleção de indivíduos. A simulação considera a escolha de uma solução migrante para um modelo constituído por 4 ilhas. Note que, o indivíduo migrante da Subpopulação 1 possui maior chance de ser selecionado para migrar à Subpopulação 4. As soluções das Subpopulações 2 e 3 apresentam, respectivamente, a segunda e terceira maior chance de seleção. Eventualmente, se o valor aleatório escolhido pelo mecanismo fosse igual à 0,5532, o indivíduo migrante da Subpopulação 2 seria selecionado.

Origem do Indivíduo Migrante	$f(x)$	Porcentagem de Seleção (ps)	Intervalo de Seleção
Subpopulação 1	1000,32	0,3933 (39,33%)	0,0 - 0,3933
Subpopulação 2	942,65	0,3706 (37,06%)	0,3934 - 0,764
Subpopulação 3	600,21	0,2361 (23,61%)	0,765 - 1,0
Total	2543,18	1,0 (100,00%)	0,0 - 1,0

Tabela 4.1: Medidas de aptidão das possíveis escolhas da *Subpopulação 4* em um modelo com 4 ilhas.

Esta topologia se destaca pelo uso do método de seleção por roleta. O privilégio concedido aos melhores indivíduos do IM é capaz de aumentar a pressão seletiva exercida. Este aumento pode acelerar a convergência do modelo. Em alguns problemas, este fator pode garantir boas soluções em tempos computacionais menores. Em outros, o aumento da pressão seletiva pode diminuir a diversidade e conduzir o modelo à uma convergência prematura.

A escolha de indivíduos de migrantes mais aptos não garante a evolução das ilhas no processo migratório. Este efeito pode ser comprovado em migrações onde duas ou mais ilhas estejam concentradas em uma mesma região do espaço de buscas. Além disto, esta topologia não mantém um histórico de suas decisões e seus respectivos ganhos. Esta característica impossibilita a detecção dos fluxos que mais contribuem para evolução das subpopulações do IM.

4.2.3 Seleção por Torneio

Esta topologia de migração dinâmica é baseada no método de *seleção por torneio*. Este método é um processo probabilístico onde indivíduos são selecionados através de pequenos “torneios” (Goldberg & Deb 1991). Os torneios são sempre realizados entre pares de indivíduos escolhidos aleatoriamente. O vencedor do torneio participa em outra competição contra uma nova solução também selecionada de maneira aleatória. Este processo realiza-se até que o número de torneios (nt) seja atendido. O parâmetro nt é responsável por controlar a pressão seletiva durante a seleção dos indivíduos.

Para a realização do torneio, o parâmetro $kt = \{kt \in \mathbb{R} \mid 0 \leq kt \leq 1\}$ deve ser configurado. Este parâmetro é utilizado na seleção de indivíduos durante as disputas. A competição acontece sorteando um número aleatório que será comparado ao parâmetro kt . Caso o valor sorteado seja menor que kt , a solução com maior aptidão é a vencedora, caso contrário, a outra solução ganha a disputa. O Algoritmo 4 descreve o funcionamento deste método, onde $nt = 2$ e $kt = 0,75$.

Assim como as outras topologias dinâmicas apresentadas nesta seção, esta abordagem representa o IM como um grafo completo. Seu fluxo migratório emerge por meio da aplicação da seleção por torneio para cada ilha do modelo. Esta seleção define quais as soluções que cada subpopulação receberá. Soluções pertencentes a própria ilha que seleciona os indivíduos migrantes não participam desta competição.

O mecanismo utilizado nesta estratégia é de fácil implementação e permite controlar a

Algoritmo 4: Algoritmo de Seleção por Torneio

```

1  $kt \leftarrow 0.75$ ;
2  $rand \leftarrow$  valor aleatório entre  $[0, 1]$ ;
3 Escolhe dois indivíduos aleatoriamente;
4 if  $rand < kt$  then
5   | Escolhe o melhor indivíduo;
6 else
7   | Escolhe o pior indivíduo;
8 Retornar o indivíduo selecionado;
```

pressão seletiva exercida. Estes fatores tornam esta topologia de migração mais atraente que as demais topologias baseadas em métodos de seleção já discutidas.

Apesar de suas boas características, estas topologias de migração consideram apenas a aptidão das soluções migrantes. Esta condição limita o controle da diversidade populacional no IM. Suas decisões baseadas em fatos momentâneos ignoram o histórico evolutivo das subpopulações. Além disto, este mecanismo necessita que alguns parâmetros do método de seleção sejam configurados.

4.3 Topologias baseadas em Métodos de Recompensa

4.3.1 Grafo Completo - Recompensa Simples

Esta abordagem de topologia dinâmica baseada em um mecanismo de recompensa foi proposta por Lardeux & Goëffon (2010). A topologia proposta pelos autores representa o IM como um grafo completo. Os nodos deste grafo descrevem as ilhas do modelo e as arestas as possíveis migrações entre elas. Nesta estratégia, cada aresta apresenta uma probabilidade de utilização de seu fluxo migratório (pa_m), conforme a Figura 4.2.

Este estudo considera que não existem fluxos migratórios para a própria subpopulação. Apenas são possíveis arestas entre ilhas distintas. Os valores das probabilidades de migração em cada aresta são dados por $pa_m = \{pa_m \in \mathbb{R} \mid 0 \leq pa_m \leq 1\}$. Seus valores iniciais são igualmente distribuídos entre todas as arestas que apresentam a mesma subpopulação como ilha de origem, de acordo com a seguinte equação:

$$pa_m = \frac{1}{\text{NUM_ILHAS} - 1} \quad (4.2)$$

onde NUM_ILHAS representa o número de subpopulações do modelo projetado.

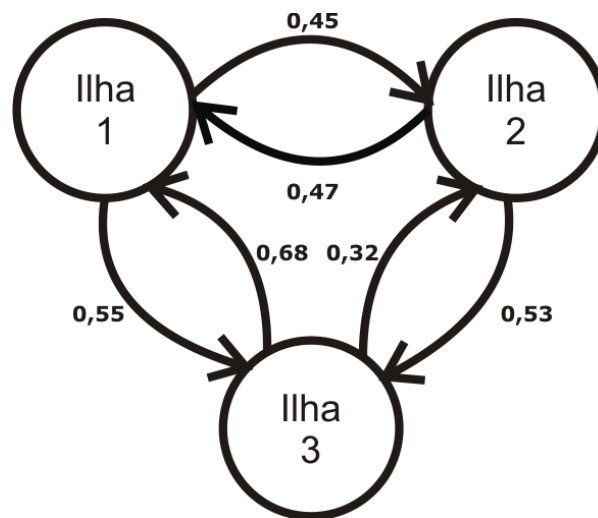


Figura 4.2: Representação do grafo completo para o IM definido por 3 subpopulações.

O processo de configuração do fluxo migratório é realizado por ilha do modelo. Cada ilha define seu fluxo local momentâneo através do método de seleção por roleta (Jaddan, Rajamani & Rao 2005). A aptidão de cada aresta é dada por sua probabilidade de utilização. Arestas com maiores probabilidades indicam bons níveis de colaboração entre as ilhas e possuem mais chances de serem novamente selecionadas.

O aprendizado da topologia ideal acontece por meio de um simples mecanismo de recompensa que atualiza as probabilidades das arestas do grafo. Esta atualização é realizada em função do efeito causado pelo fluxo migratório aplicado. Fluxos que melhoram a evolução das subpopulações são recompensados. Fluxos cujas as soluções migrantes não contribuíram para evolução das ilhas às quais foram destinados são penalizados. Este procedimento se realiza acrescentando ou diminuindo $rpa_m = \{rpa_m \in \mathbb{R} \mid 0 \leq rpa_m \leq 1\}$ às probabilidades das arestas, de acordo com sua contribuição.

A melhora na evolução das ilhas pode ser identificada por diferentes *indicadores*. Em geral, estes indicadores avaliam a diversidade populacional e/ou a medida de custo dos

indivíduos das subpopulações. Eles tomam como base o estágio anterior ao processo de migração e o estado logo após a ele. Estes indicadores funcionam como oráculo ao processo de aprendizado. Os indicadores avaliados neste trabalho são apresentados na subseção 4.3.4.

Esta topologia de migração é capaz de detectar rapidamente os melhores fluxos migratórios. Isto é possível devido à simplicidade do modelo de aprendizado proposto. Entretanto, a definição do melhor indicador e do parâmetro do método de recompensa não é uma tarefa trivial. Em um ambiente altamente dinâmico como o IM, a compreensão de novos estágios por esta estratégia pode ser demorada. Assim, a reinicialização das probabilidades das arestas do grafo pode se fazer necessária.

4.3.2 Sistema Multiagentes - Q-learning

Esta topologia de migração dinâmica transforma o IM em um Sistema Multiagentes (MAS) (Lopes, Silva, Campelo & Guimaraes 2012). Neste sistema, cada subpopulação do modelo é associada a um agente. A principal função do agente é aprender os fluxos que mais contribuem para evolução de sua ilha associada. Eles atuam durante todo o processo evolucionário de forma independente.

Os agentes projetados para esta estratégia adquirem conhecimento por meio da experiência obtida de processos migratórios anteriores. Este processo de aprendizado é baseado no algoritmo *Q-learning* (Watkins & Dayan 1992). O algoritmo Q-learning é um tradicional método estocástico de aprendizado por reforço. Seu modelo de aprendizado é realizado por meio de interações de *tentativa-e-erro* com o ambiente (Kaelbling, Littman & Moore 1996).

Cada agente desta estratégia é composto por estados e ações. Os estados descrevem as ilhas a partir dos quais um agente definido pode receber indivíduos no processo migratório. Cada estado do agente é associado a uma única subpopulação do modelo. As ações representam a troca ou manutenção da subpopulação escolhida para enviar soluções. Assim, o número de estados (NUM_ESTADOS) e o número de possíveis ações (NUM_ACOES) dos agentes são determinadas pelas respectivas equações:

$$\text{NUM_ESTADOS} = (\text{NUM_ILHAS} - 1) \quad (4.3)$$

$$\text{NUM_ACOES} = (\text{NUM_ESTADOS})^2 \quad (4.4)$$

onde NUM_ILHAS representa o número de subpopulações do modelo.

A Figura 4.3 ilustra o funcionamento destes agentes. Neste diagrama, as ilhas são subpopulações e as arestas azuis são estados alcançáveis pelo Agente 4. A aresta sólida azul representa o estado atual e significa que a Ilha 4 está recebendo indivíduos da Ilha 2. As arestas vermelhas representam as possíveis ações do Agente 4 no estado atual. Se o Agente 4 escolhe a Ação 1, a Ilha 4 continuará recebendo indivíduos da Ilha 2; se, por outro lado, o Agente 4 escolhe a Ação 3, a Ilha 4 trocará seu fluxo migratório interrompendo a entrada de indivíduos da Ilha 2 e começando a receber soluções da Ilha 1. A operação dos demais agentes funciona da mesma forma.

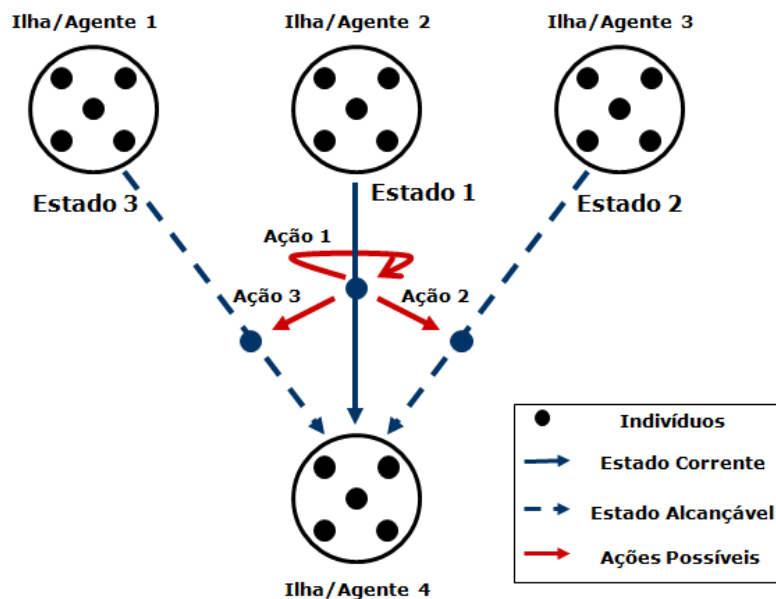


Figura 4.3: Representação de um agente para IM constituído por 4 ilhas.

Os agentes deste sistema escolhem suas ações por si só e são recompensados ou punidos por elas. Desta maneira, o processo de aprendizado emerge da maximização das recompensas obtidas por parte do agente. O aprendizado das melhores ações em cada estado é possível após a aplicação de uma sequência de ações. Ela acontece observando a recompensa dada à ação realizada. Assim, este método aprende a política

ótima de ações usando a seguinte equação:

$$Q(s_t, a_t) \leftarrow \lambda(r + \gamma \max_{\alpha'} Q(s_{t+1}, a_{t+1})) + (1 - \lambda)Q(s_t, a_t) \quad (4.5)$$

onde, s_t é o estado atual; a_t é ação aplicada em s_t ; r é a recompensa recebida depois de aplicado a ação a_t ; s_{t+1} é o novo estado; γ é o fator de desconto; e λ é a taxa de aprendizagem.

O pseudocódigo do método Q-learning é apresentado no Algoritmo 5. Inicialmente, o método escolhe aleatoriamente o par estado-ação. Em seguida, o algoritmo busca determinar as melhores ações a partir do estado definido. Estas ações são definidas estimando recompensas através de experiências passadas e estimativas futuras.

Algoritmo 5: Método de Aprendizado por Reforço - Q-learning

```

1 ENTRADA
2  $\lambda \leftarrow$  taxa de aprendizagem;
3  $\gamma \leftarrow$  fator de desconto;
4  $\epsilon \leftarrow$  taxa de exploração;
5 INÍCIO
6  $\forall_s \forall_a Q(s, a) \leftarrow 0$ ;
7  $s_t \leftarrow$  estado corrente;
8 if  $rand() < \epsilon$  then
9   |  $a_t \leftarrow$  ação aleatória;
10 else
11   |  $a_t \leftarrow \arg \max_{\alpha} Q(s_t, a_t)$ ;
12 end
13 Aplica a ação  $a_t$ ;
14 Calcular a recompensa  $r$ ;
15  $s_{t+1} \leftarrow$  estado corrente;
16  $Q(s_t, a_t) \leftarrow \lambda(r + \gamma \max_{\alpha'} Q(s_{t+1}, a_{t+1})) + (1 - \lambda)Q(s_t, a_t)$ ;
17  $\lambda \leftarrow 0.99 * \lambda$ ;
18  $\epsilon \leftarrow 0.98 * \epsilon$ ;
19 goto 6;
20 FIM

```

O método Q-learning é composto de alguns parâmetros, como visto no algoritmo apresentado. Note que estes parâmetros podem influenciar seu desempenho. O parâmetro taxa de aprendizado determina a importância dada as novas informações sobre as antigas. O fator de desconto determina a relevância das recompensas futuras. A taxa de

exploração define se o agente decidirá por selecionar a melhor ação até o momento ou irá explorar o espaço de ações.

O processo de migração desta topologia dinâmica acontece posteriormente à definição do fluxo migratório pelos agentes. Após a migração cada agente é recompensado de acordo com os indicadores apresentados na subseção 4.3.4. Detalhes sobre o funcionamento desta topologia de migração podem ser visualizados no Algoritmo 6.

Estudos já realizados comprovam que o método de aprendizado utilizado por esta topologia de migração converge para uma política ótima de ações (Watkins & Dayan 1992). No entanto, seu modelo de aprendizado é muito complexo e requer a configuração de vários parâmetros essenciais. A reinicialização também pode ser necessária para esta estratégia devido à dinâmica do processo evolutivo.

Algoritmo 6: Topologia baseada no método Q-learning

```

1  ENTRADA
2   $\lambda \leftarrow$  taxa de aprendizagem;
3   $\gamma \leftarrow$  fator de desconto;
4   $\epsilon \leftarrow$  taxa de exploração;
5   $GEN \leftarrow$  intervalo de migração;
6  INÍCIO
7  for cada subpopulação do
8  |   Inicialize o estado inicial do agente Q-learning aleatoriamente;
9  |   Inicialize a subpopulação;
10 end
11 while  $\neg$  critério_de_parada do
12 |   for cada subpopulação do
13 |   |   Aplica o algoritmo de otimização (GEN);
14 |   end
15 |   for cada subpopulação do
16 |   |   Seleciona a ação do agente Q-learning;
17 |   |   Aplica o processo de migração;
18 |   |   Estima a recompensa;
19 |   |   Atualiza o agente Q-learning;
20 |   end
21 end
22 FIM

```

4.3.3 Sistema Multiagentes - Upper Confidence Bound

Semelhante à estratégia apresentada na subseção anterior, esta topologia de migração dinâmica transforma o IM em um MAS. A representação deste modelo é conduzida da mesma maneira. Assim, cada agente do sistema representa uma única subpopulação do modelo. Estes agentes são capazes de aprender e determinar os melhores fluxos migratórios momentâneos.

O processo de aprendizado é realizado nesta topologia pelo algoritmo *Upper Confidence Bound* (UCB) (Auer, Cesa-Bianchi & Fischer 2002). Assim como o Q-learning, este método de aprendizado por reforço também foi proposto no contexto de Teoria de Jogos. Ele já foi aplicado com sucesso na resolução de problemas como *Multi-armed Bandit* (Maturana, Fialho, Saubion, Schoenauer & Sebag 2009) e *Random-armed Bandit* (Auer 2003).

O problema de configuração da topologia de migração é adaptado por esta proposta ao problema *Multi-armed Bandit* (MAB). Assim, cada agente do sistema é composto por vários “braços”. Estes braços descrevem subpopulações do IM capazes de enviar indivíduos migrantes à ilha associada ao agente. A quantidade de braços em cada agente é determinada por:

$$\text{NUM_BRACOS} = (\text{NUM_ILHAS} - 1) \quad (4.6)$$

onde NUM_ILHAS descreve o número de subpopulações que compõem o modelo.

O Algoritmo 7 descreve o método de aprendizado por reforço UCB. Em sua etapa inicial todos os braços são selecionados uma vez pelo agente. Os braços selecionados são recompensados ou punidos de acordo com indicadores descritos na subseção 4.3.4. Após a etapa inicial, o algoritmo UCB seleciona, a cada processo de migração, o braço que maximize a seguinte equação (Maturana, Fialho, Saubion, Schoenauer & Sebag 2009):

$$\text{rb} = \hat{p}_{i,t} + C \sqrt{\frac{\log \sum_{kb=1}^{\text{NUM_BRACOS}} n_{kb,t}}{n_{i,t}}} \quad (4.7)$$

onde, $\hat{p}_{i,t}$ é a média da recompensas recebidas pelo braço i até o tempo t ; C é um fator

de escala; e $n_{i,t}$ é a quantidade de vezes que o braço i foi selecionado até o tempo t .

Na equação 4.7, o primeiro termo favorece os melhores braços, ou seja, braços que foram ao longo do tempo mais bem recompensados. O segundo garante que cada braço seja selecionado infinitas vezes pelo método (Maturana, Fialho, Saubion, Schoenauer & Sebag 2009). Desta forma, o fator de escala C inserido na fórmula visa equilibrar os termos da equação.

O funcionamento completo desta topologia de migração dinâmica é apresentado no Algoritmo 8. Nesta topologia, em um primeiro instante os agentes definem a topologia de migração momentânea do modelo. Após o processo migratório as recompensas de cada agente são calculadas e repassadas à eles. O processo de evolução do IM continua até que o critério de parada seja atendido.

O modelo de aprendizado utilizado nesta abordagem é muito mais simples comparado ao modelo do algoritmo Q-learning. Este modelo garante a escolha de todos os fluxos por pelo menos uma rodada. Porém, existe a necessidade de configuração e reinicialização dos parâmetros deste mecanismo.

Algoritmo 7: Método de Aprendizado por Reforço - UCB

```

1 ENTRADA
2  $C \leftarrow$  fator de escala;
3 NUM_BRACOS  $\leftarrow$  número de braços;
4  $CONT \leftarrow 0$ ;
5 INÍCIO
6 while  $\neg$  critério_de_parada do
7    $CONT \leftarrow CONT + 1$ ;
8   if  $CONT \leq NUM\_BRACOS$  then
9     | Selecione um braço ainda não utilizado;
10  else
11    | for cada braço do
12      | Calcular  $rb$ ;
13    | end
14    | Selecione o braço com maior valor de  $rb$ ;
15  end
16  Aplica o braço selecionado;
17  Receba a recompensa  $r$ ;
18 end
19 FIM

```

Algoritmo 8: Topologia baseada no método UCB

```

1 ENTRADA
2  $C \leftarrow$  fator de escala;
3  $GEN \leftarrow$  intervalo de migração;
4 INÍCIO
5 for cada subpopulação do
6   | Inicialize a subpopulação;
7 end
8 while  $\neg$  critério_de_parada do
9   | for cada subpopulação do
10  |   | Aplica o algoritmo de otimização (GEN);
11  |   end
12  |   for cada subpopulação do
13  |   | Seleciona o braço do agente;
14  |   | Aplica o processo de migração;
15  |   | Estima a recompensa;
16  |   | Atualiza os valores do braço agente;
17  |   end
18 end
19 FIM

```

4.3.4 Indicadores Populacionais

Melhores indivíduos

Este indicador considera a aptidão das melhores soluções das subpopulações envolvidas no processo de migração. Sua fórmula é definida por:

$$ID1 = \frac{(APT_CORRENTE - APT_RECEBIDA)}{\max(APT_CORRENTE, APT_RECEBIDA)} \quad (4.8)$$

onde APT_CORRENTE é o valor da função de custo do melhor indivíduo da subpopulação corrente e APT_RECEBIDA é o valor da função de custo da solução recebida no processo de migração.

Medida de custo média

Medida de custo média representa a média da função de custo dos indivíduos das subpopulações do IM antes e depois do processo de migração. Este indicador é definido como:

$$ID2 = \frac{(MEDIA_CORRENTE - MEDIA_ANTERIOR)}{\max(MEDIA_CORRENTE, MEDIA_ANTERIOR)} \quad (4.9)$$

onde $MEDIA_CORRENTE$ é média da função de custo dos indivíduos da subpopulação depois do processo de migração e $MEDIA_ANTERIOR$ é a média da função de custo antes da realização do processo de migração.

Diversidade populacional

O indicador da diversidade populacional toma como base as diversidades das subpopulações antes e depois ao processo de migração. Este indicador é descrito pela seguinte equação:

$$ID3 = \frac{(DIVERSIDADE_CORRENTE - DIVERSIDADE_ANTERIOR)}{\max(DIVERSIDADE_CORRENTE, DIVERSIDADE_ANTERIOR)} \quad (4.10)$$

onde $DIVERSIDADE_CORRENTE$ é a média da distância euclidiana entre as soluções da subpopulação após a migração e $DIVERSIDADE_ANTERIOR$ é a distância média entre as soluções antes ao processo de migração.

Medida de custo média \times Diversidade populacional

Este indicador avalia simultaneamente dois outros indicadores já apresentados. Ele considerada a ponderação entre os mesmos, como apresenta a equação a seguir:

$$ID4 = ID2 \times (1 - fp) + ID3 \times FP \quad (4.11)$$

onde ID2 é o indicador da medida de custo média; ID3 é o indicador da diversidade populacional; $fp = \{fp \in \mathbb{R} \mid 0 \leq fp \leq 1\}$ e representa a ponderação entre os indicadores.

4.4 Topologias baseadas em Técnicas de Agrupamento

4.4.1 K-medoids

Esta topologia de migração foi proposta por Berntsson & Tang (2005). Seu mecanismo de configuração é baseado em técnicas de agrupamento. Estas técnicas de agrupamento são classificadas como métodos de aprendizado não-supervisionado. Basicamente, elas dividem automaticamente os dados em grupos com base na similaridade (Tan, Steinbach & Kumar 2005).

O processo que define o fluxo migratório para esta topologia é composto por duas etapas. A primeira etapa determina os grupos de soluções das subpopulações do IM, empregando o algoritmo *K-medoids* (Kaufman & Rousseeuw 1987). Este processo considera o melhor indivíduo de cada ilha do modelo. Ilhas que foram definidas em grupos unitários não participam do processo de migração. A segunda etapa configura as topologias para os grupos definidos. Em cada grupo, topologias em anel ou aleatória são construídas entre as ilhas. Este procedimento se repete a cada processo migratório realizado durante a execução do IM.

O algoritmo K-medoids foi proposto ao final dos anos 80 por Kaufman & Rousseeuw (1987). O objetivo deste método é particionar um conjunto de n dados em k grupos. Este método busca definir grupos que minimizam a distância entre os dados a cerca dos k pontos mais representativos. Ele tem sido aplicado com sucesso em diversos trabalhos encontrados na literatura (Krishnapuram, Joshi & Yi 1999), (Berntsson & Tang 2005), (Garcia-Baleon, Alarcon-Aquino & Starostenko 2009).

Durante o processo de definição dos grupos, o algoritmo K-medoids requer o cálculo da distância entre soluções distintas. Para determinar esta distância, este estudo considera duas diferentes métricas. As métricas avaliadas neste trabalho são descritas na subseção 4.4.2. Assim, o Algoritmo 9 apresenta detalhes do funcionamento desta topologia de migração dinâmica.

Esta topologia de migração garante a troca de informações entre subpopulações com o mesmo nível de desenvolvimento. No entanto, o algoritmo de agrupamento K-medoids

demanda processamento e este fator se eleva como o aumento dos dados. A definição do número de grupos considerados é outro fator complicador.

Algoritmo 9: Topologia baseada na Técnica de Agrupamento K-medoids

```

1 ENTRADA
2  $k \leftarrow$  número de grupos,  $GEN \leftarrow$  intervalo de migração;
3 INÍCIO
4 for cada subpopulação do
5   | Inicialize a subpopulação;
6 end
7 while  $\neg$  critério_de_parada do
8   | for cada subpopulação do
9     | Aplica o algoritmo de otimização (GEN);
10  | end
11  | Defina  $k$  grupos de soluções;
12  | for cada grupo do
13    | Defina a topologia de migração;
14    | Aplica o processo de migração;
15  | end
16 end
17 FIM

```

4.4.2 Métricas de distância

Distância Euclidiana

Para dois pontos n -dimensionais, $P = [p_1, p_2, \dots, p_n]$ e $Q = [q_1, q_2, \dots, q_n]$, a distância é calculada como:

$$D1 = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (4.12)$$

Distância de Manhattan

Para dois pontos n -dimensionais, $P = [p_1, p_2, \dots, p_n]$ e $Q = [q_1, q_2, \dots, q_n]$, a distância é calculada como:

$$D2 = \sum_{i=1}^n \|p_i - q_i\| \quad (4.13)$$

Capítulo 5

Resultados

Este capítulo apresenta os experimentos computacionais realizados utilizando as topologias de migração dinâmicas estudadas no Capítulo 4. Os desempenhos detas são comparadas tanto entre si quanto com as tradicionais topologias estáticas em anel e estrela. Assim, as seções subsequentes descrevem detalhes dos experimentos, resultados e a análise estatística conduzida.

5.1 Configuração dos Testes

5.1.1 Método de Otimização

Para comparação das diferentes versões do IM, o método de otimização DE foi implementado empregando cada uma das topologias estudadas. Os parâmetros do DE utilizados nos experimentos foram determinados por meio de testes preliminares e são apresentados na Tabela 5.1. Cada configuração foi executada independentemente 15 vezes para cada problema de otimização avaliado. O processo de busca era finalizado quando um dos critérios descritos a seguir fosse satisfeito:

1. Número máximo de 20000 gerações tenha sido atingido;
2. $\Delta f \leq 0.0001$ é encontrado¹.

¹ Δf é o valor de precisão do objetivo. Ele é definido por $\Delta f = f_{atingido} - f_{ótimo}$

Parâmetro	Valor Definido
F	0,7
CR	0,9
NP	100 indivíduos por ilha do modelo
Operador de Mutação Diferencial	$rand/1$

Tabela 5.1: Parâmetros do algoritmo DE utilizados nos experimentos.

5.1.2 Problemas de Otimização

Nestes experimentos, 6 funções de testes do Conjunto de Funções para Otimização Global em Larga Escala (Tang, Yao, Suganthan, MacNish, Chen, Chen & Yang 2007) foram avaliadas. Estas funções são descritas no Apêndice A.1. As funções f_1 e f_2 são unimodais e as funções f_3 , f_4 , f_5 e f_6 são multimodais. Todas elas foram definidas neste estudo com 100 variáveis.

5.1.3 Parâmetros - Modelo em Ilhas

O Modelo em Ilhas utilizado nos experimentos foi implementado em arquitetura sequencial. Nestas versões do IM implemetadas, os melhores indivíduos de cada subpopulação migram e substituem indivíduos aleatórios da subpopulação que os recebem. O processo de migração ocorre a cada 20 gerações e cada ilha do modelo recebe apenas um indivíduo por migração. Modelos compostos por 4, 8, 16 e 32 subpopulações foram considerados para este experimento. Estes parâmetros foram definidos através de pequenos experimentos prévios realizados e relatos de trabalhos anteriores (Ruciński, Izzo & Biscani 2010), (Lopes, Silva, Campelo & Guimaraes 2012).

5.1.4 Nomenclatura das Topologias Estudadas

Por simplicidade, a nomenclatura apresentada a seguir foi definida para melhor identificação das topologias de migração estudadas:

- Aleatório: versão do IM utilizando a topologia de migração dinâmica baseada na seleção aleatória;

- Anel: versão do IM utilizando a topologia de migração estática em anel;
- Estrela: versão do IM utilizando a topologia de migração estática em estrela;
- Grafo Completo: versão do IM utilizando a topologia de migração dinâmica baseada na recompensa das arestas do grafo;
- K-medoids: versão do IM utilizando a topologia de migração dinâmica baseada na técnica de agrupamento K-medoids;
- Q-learning: versão do IM utilizando a topologia de migração dinâmica baseada no método de aprendizado Q-learning;
- Roleta: versão do IM utilizando a topologia de migração dinâmica baseada no método de seleção por roleta;
- Torneio: versão do IM utilizando a topologia de migração dinâmica baseada no método de seleção por torneio;
- UCB: versão do IM utilizando a topologia de migração dinâmica baseada no método de aprendizado UCB;

5.1.5 Testes Estatísticos

Para validação estatística dos resultados, a análise de variância baseada em permutação (ANOVA) (Cochran, Cochran & Cox 1992) foi utilizada devido à não-normalidade dos dados. Quando diferenças significativas foram detectadas para as topologias, testes permutacionais (pacote *coin* do software estatístico R) foram utilizados para as comparações post-hoc e os valores-p obtidos durante as comparações foram ajustados segundo o método de Benjamini & Hochberg (1995). O modelo estatístico considerado para os dados foi um planejamento fatorial (Montgomery 2006), onde as topologias, o número de ilhas e os problemas testes são fatores experimentais. Além disso, o valor α foi previamente definido como 0.05.

5.2 Resultados Comparativos

5.2.1 Topologia Aleatória

Esta seção apresenta os resultados comparativos da abordagem multipopulacional (IM) empregando as topologias Aleatória, Anel e Estrela. A Figura 5.1 apresenta a distribuição da taxa de convergência acumulada do tempo de execução² para as diferentes topologias. IM compostos por 4, 8, 16 e 32 ilhas foram avaliados neste experimento e os resultados demonstram similaridade entre as taxas de convergência das diferentes estratégias de migração. Nota-se que a topologia Aleatória teve seu desempenho degradado com o aumento do número de ilhas, sendo superada na maioria das configurações pela topologia Anel. Apesar da perda de desempenho, a topologia Aleatória foi superior à Estrela em todas as configurações.

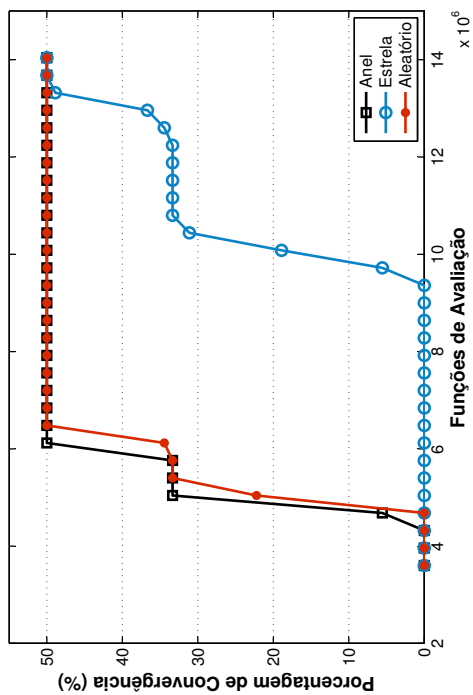
A análise de variância aplicada aos resultados das topologias estudadas indicam diferenças significativas entre as médias. Posteriormente, uma análise comparativa post-hoc foi aplicada e os resultados considerando as médias do número de funções de avaliação e qualidade da solução (Δf) são apresentados nas Tabelas 5.2 e 5.3.

Considerando 95% de confiança, os resultados apresentados na Tabela 5.2 indicam que o desempenho médio da topologia Aleatória foi significamente melhor que a Estrela. No entanto, a topologia Aleatória comparada a Anel não indica diferenças de desempenho. O desempenho médio em relação a qualidade da solução final apresentado na Tabela 5.3 revela que as topologias não apresentam diferenças significativas de desempenho médio.

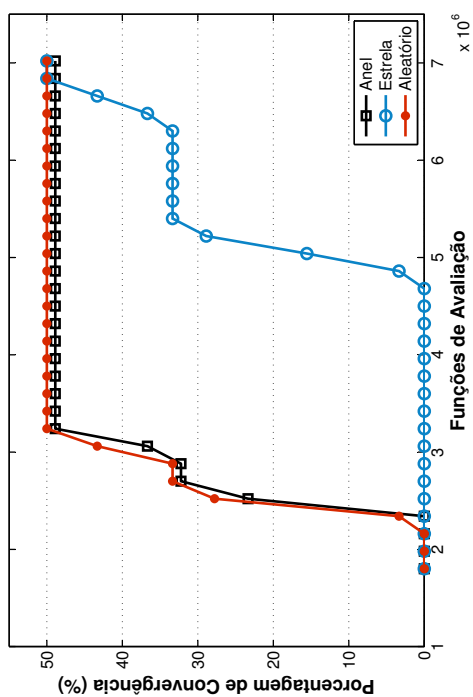
Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Aleatória \times Anel	20181911 \times 19756067	0.7267225
Aleatória \times Estrela	20181911 \times 25247133	1.433447 * 10^{-7}

Tabela 5.2: Comparação entre médias (número de funções de avaliação) em relação à topologia Aleatória.

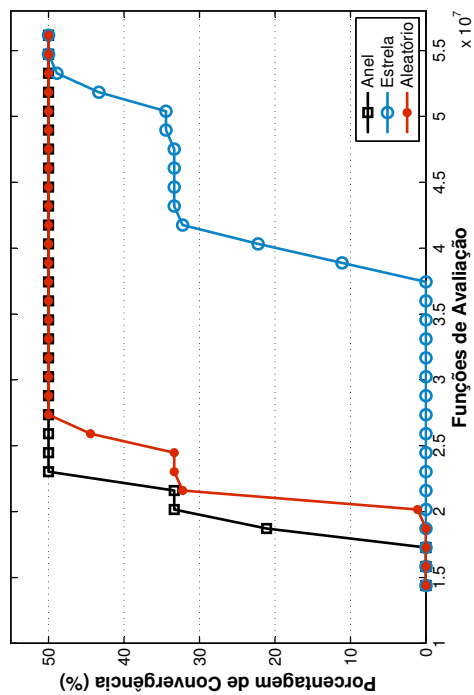
²Tempo de execução é definido como o número de avaliações da função de custo até que a solução ideal (com determinada precisão) seja atendida.



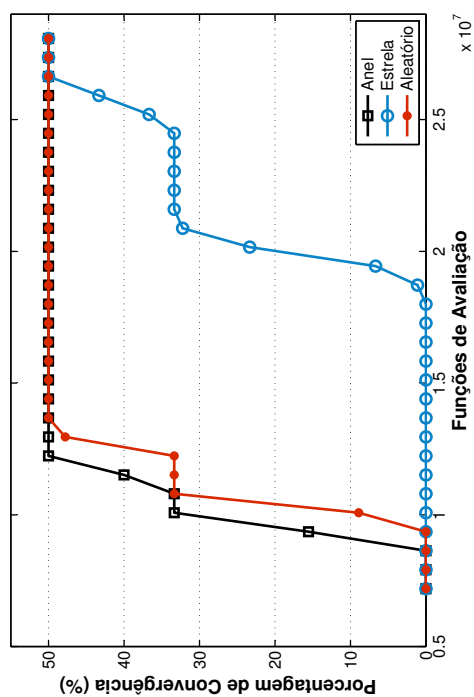
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.1: Taxa de Convergência Acumulativa - Topologia Aleatória.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Aleatória \times Anel	33.72069 \times 31.52867	0.5290177
Aleatória \times Estrela	33.72069 \times 39.83329	0.2368713

Tabela 5.3: Comparação entre médias (melhores soluções) em relação à topologia Aleatória.

5.2.2 Topologia Roleta

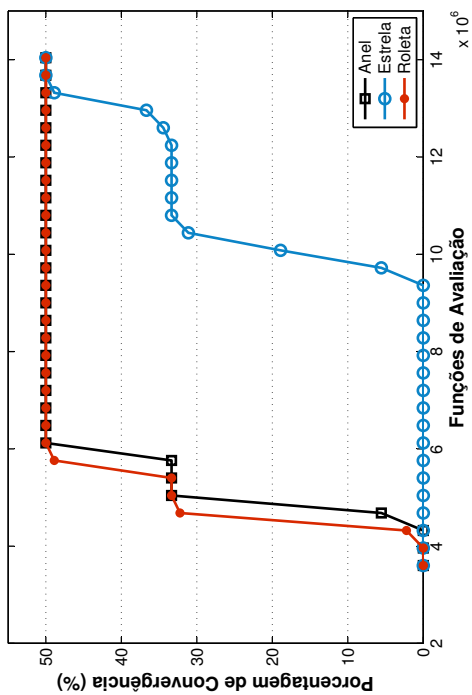
Os resultados descritos nesta seção são um comparativo do desempenho da topologia dinâmica Roleta e as topologias estáticas Anel e Estrela. A distribuição da taxa de convergência acumulada do tempo de execução para as diferentes topologias utilizando IM compostos por 4, 8, 16 e 32 ilhas é apresentada na Figura 5.2. A taxa final de convergência da topologia Anel mostrou-se inferior as demais somente em modelos com 4 ilhas. Apesar de apresentar taxas de convergência semelhantes, o IM usando a topologia Roleta foi capaz de determinar boas soluções gastando menos funções de avaliação.

Os testes estatísticos realizados corroboram a refutação da hipótese de igualdade entre as médias do IM utilizando as topologias de migração avaliadas. As Tabelas 5.4 e 5.5 representam os resultados da análise post-hoc considerando o número de funções de avaliação e a qualidade da soluções encontradas, respectivamente.

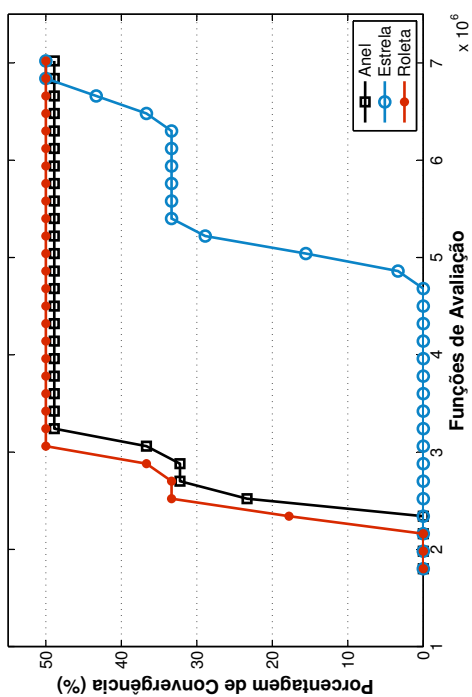
Como reportado na Tabela 5.4, o desempenho médio da topologia Roleta, considerando o número de funções de avaliação, é significativamente melhor que a Estrela e igual à Anel. Em relação a Δf (qualidade da solução encontrada), os testes sugerem que não existem diferenças significativas entre as topologias estudadas.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Roleta \times Anel	19461356 \times 19756067	0.6393583
Roleta \times Estrela	19461356 \times 25247133	2.636943 * 10^{-9}

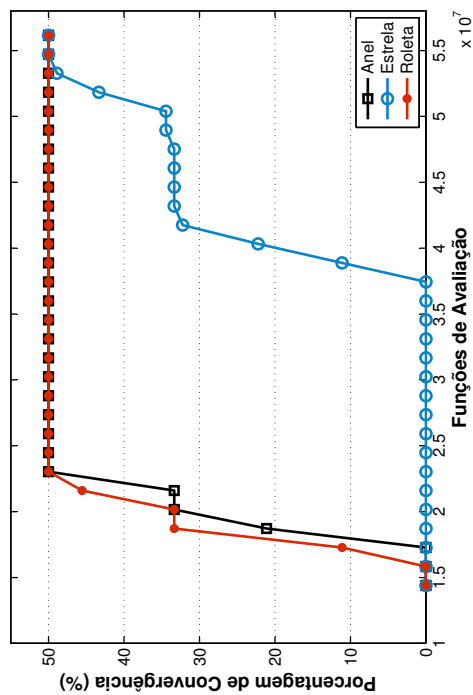
Tabela 5.4: Comparação entre médias (número de funções de avaliação) em relação à topologia Roleta.



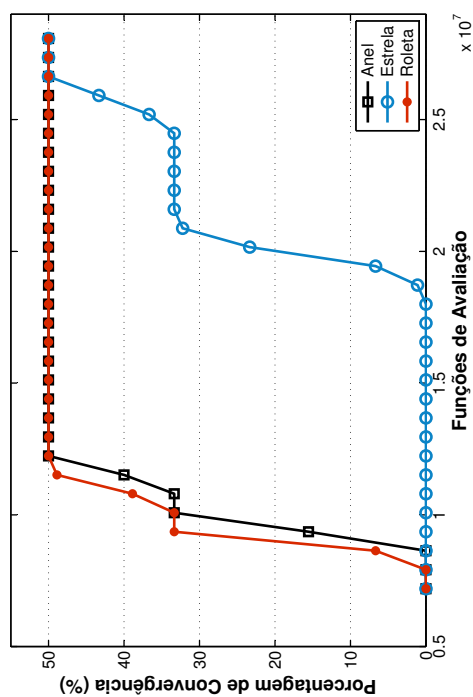
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.2: Taxa de Convergência Acumulativa - Topologia Roleta.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Roleta \times Anel	33.28293 \times 31.52867	0.6150848
Roleta \times Estrela	33.28293 \times 39.83329	0.1896848

Tabela 5.5: Comparação entre médias (melhores soluções) em relação à topologia Roleta.

5.2.3 Topologia Torneio

Nesta seção, os resultados do IM utilizando a topologia Torneio são descritos e comparados as topologias Anel e Estrela. Como estudado no Capítulo 4, o mecanismo de configuração desta topologia requer que alguns parâmetros sejam definidos previamente. Devido a influência destes parâmetros sobre o desempenho da abordagem estudada, diversos valores para os mesmos foram avaliados, vide a Tabela 5.6.

Todas as combinações do conjunto apresentado foram testadas e os experimentos indicam que torneios de tamanho 3 e altos valores para o parâmetro kt apresentam resultados mais interessantes. Os valores mais altos de kt ($kt \in [0.8; 0.9; 1.0]$) não apresentaram diferenças significativas. Assim, apenas a topologia Torneio cujo tamanho do torneio é 3 e $kt = 0.8$ será apresentada durante o restante deste trabalho.

Na Figura 5.3, a distribuição da taxa de convergência das topologias avaliadas é apresentada para modelos com 4, 8, 16 e 32 ilhas. Em sua maioria, as taxas de convergência das topologias apresentadas foram similares. É notório que, em modelos com 4, 8 e 16 ilhas, a topologia Torneio foi capaz de alcançar boas soluções consumindo menores quantidades de funções de avaliação.

O resultado da análise de variância entre as médias do IM utilizando as topologias Aleatório, Anel e Torneio sugerem que as mesmas são significativamente diferentes. Deste modo, uma análise post-hoc entre as topologias foi realizada e seu resultado é resumido nas Tabelas 5.7 e 5.8. Estas tabelas descrevem a média e o valor- p das comparações realizadas para o número de funções de avaliação e Δf .

A análise de desempenho médio do número de funções de avaliação apresentada na Tabela 5.7 indica que a média atingida pela topologia Torneio é igual à Anel e superior à Estrela. Em relação à qualidade das soluções encontradas, vide Tabela 5.8, a análise estatística sugere igualdade entre as topologias.

Parâmetro	Valores
NT	[2; 3]
kt	[0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1.0]

Tabela 5.6: Parâmetros avaliados para a topologia Torneio.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Torneio \times Anel	19665444 \times 19756067	0.8725205
Torneio \times Estrela	19665444 \times 25247133	0.1280797 * 10^{-9}

Tabela 5.7: Comparação entre médias (número de funções de avaliação) em relação à topologia Torneio.

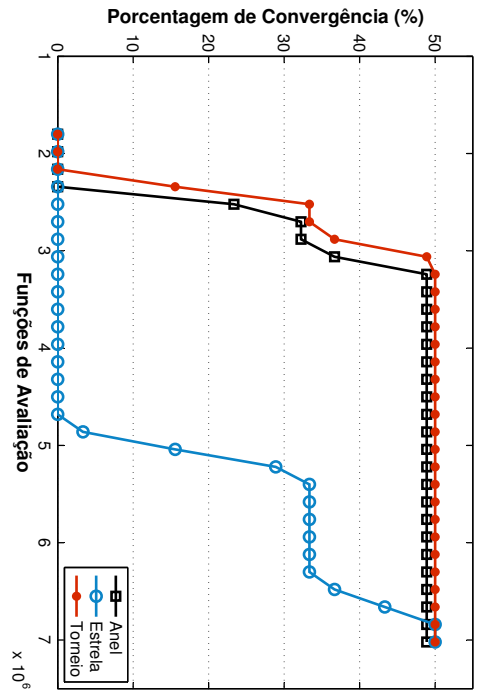
Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Torneio \times Anel	33.19843 \times 31.52867	0.7884888
Torneio \times Estrela	33.19843 \times 39.83329	0.3158926

Tabela 5.8: Comparação entre médias (melhores soluções) em relação à topologia Torneio.

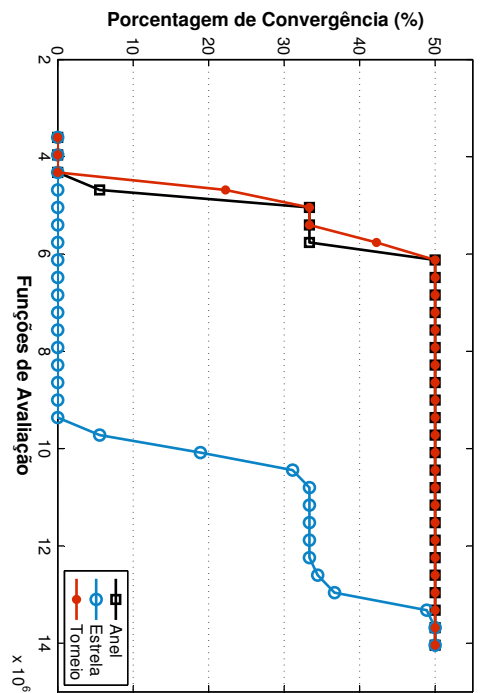
5.2.4 Topologia Grafo Completo

Para esta seção, os dados apresentados descrevem os resultados dos experimentos realizados considerando a topologia Grafo Completo. Assim como nas demais seções, esta topologia dinâmica é comparada com duas outras topologias estáticas: Anel e Estrela. A topologia Grafo Completo necessita da configuração de alguns parâmetros (pa_m , indicador de evolução e reinicialização das probabilidades de migração). Diversos valores para estes parâmetros foram avaliados e são apresentados na Tabela 5.9.

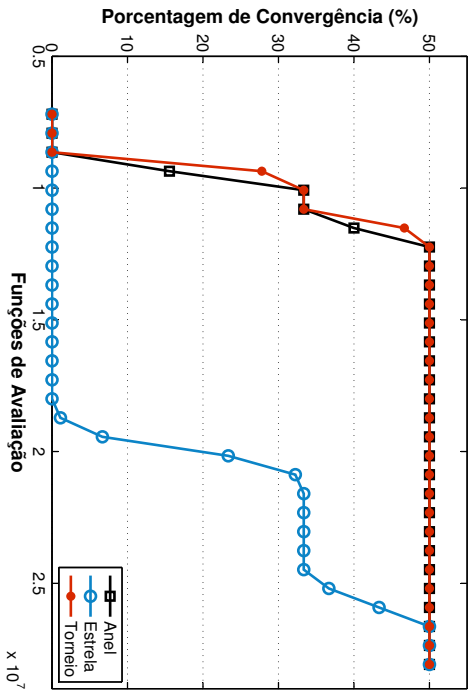
A análise estatística dos resultados das diferentes configurações avaliadas indicam que não há diferença significativa entre elas. Assim, a configuração dos parâmetros que apresentou menor média estimada (número de funções de avaliação) foi escolhida para representar esta abordagem estudada. Seus valores são definidos como: $pa_m = 0.05$, melhores indivíduos (Indicador, vide seção 4.3.4) e sem reinicialização de parâmetros.



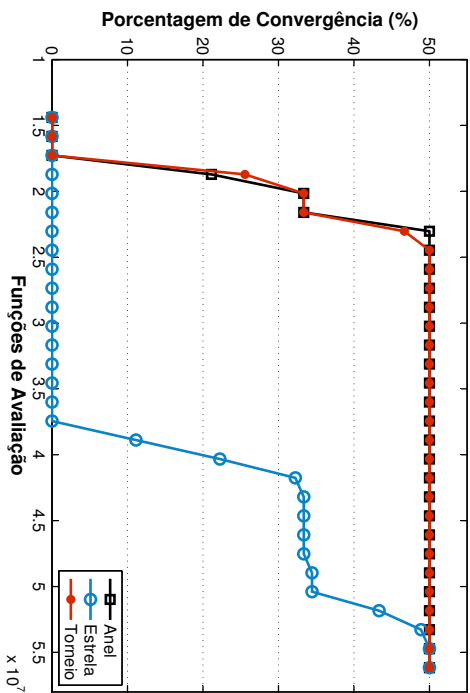
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.3: Taxa de Convergência Acumulativa - Topologia Torneio.

Parâmetro	Valores
pa_m	0.05 (Valor definido pelos autores (Lardeux & Goëffon 2010))
Indicador	[Melhores indivíduos; Medida de custo média; Diversidade populacional; Medida de custo média \times Diversidade populacional]
fp (Parâmetro do Indicador Medida de custo média \times Diversidade populacional)	[0.3; 0.5; 0.7]
Reinicialização	[Nunca; a cada 50 gerações; a cada 100 gerações]

Tabela 5.9: Parâmetros avaliados para a topologia Grafo Completo.

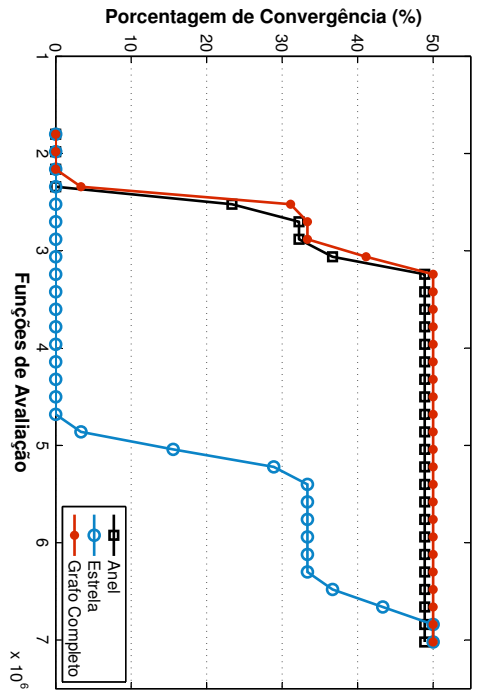
A Figura 5.4 compara a distribuição da taxa de convergência entre as diferentes topologias de migração, levando em conta modelos com 4, 8, 16 e 32 ilhas. Os gráficos demonstram que em modelo iguais ou superiores à 8 ilhas, a taxa final de convergência das topologias avaliadas são iguais. A topologia Grafo Completo tende a ter seu desempenho pior que a topologia Anel com o aumento do número de ilhas do modelo. Mas em comparação a topologia Estrela, ela foi superior em todas as configurações estudadas.

Devido à refutação da hipótese de igualdade entre as médias atingidas pelo IM utilizando as diferentes topologias de migração, a análise post-hoc foi realizada. Os resultados desta análise são descritos nas Tabelas 5.10 e 5.11. Estas tabelas apresentam um comparativo entre as topologias estudadas nesta seção, considerando o número de funções de avaliação e qualidade das soluções encontradas.

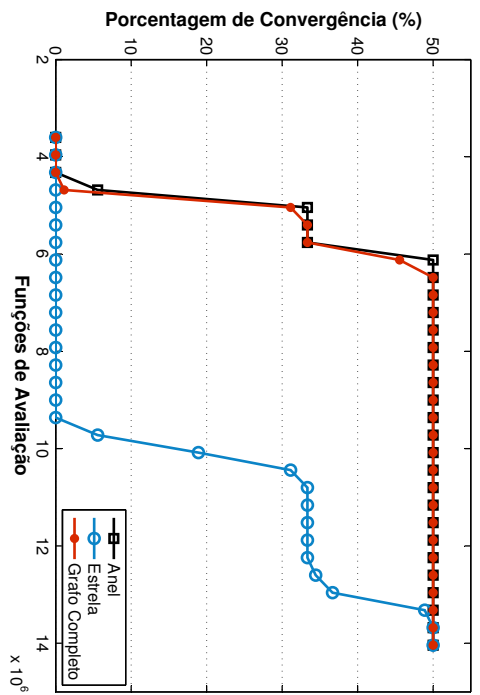
Na Tabela 5.10, conclui-se com 95% de confiança que a topologia Grafo Completo foi superior à Estrela e similar à topologia Anel. Em relação aos resultados da variável de resposta Δf , apresentados na Tabela 5.11, os testes indicam que não existem diferenças significativas entre as abordagens.

5.2.5 Topologia Qlearning

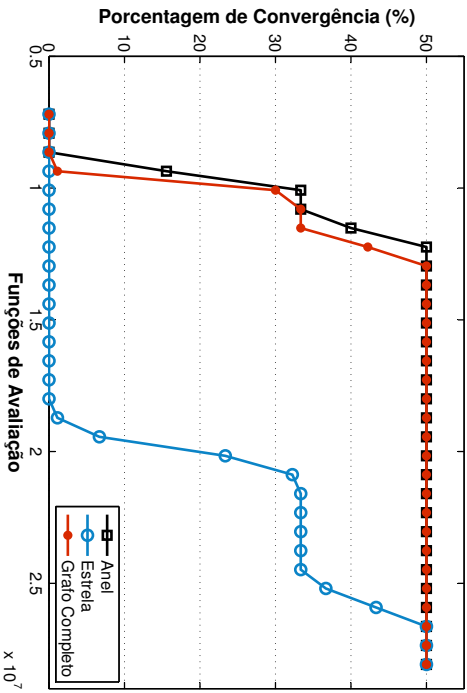
Esta seção apresenta os resultados comparativos do IM utilizando as topologias Anel, Estrela e Q-learning, descritas nos capítulos anteriores. A topologia dinâmica Q-learning



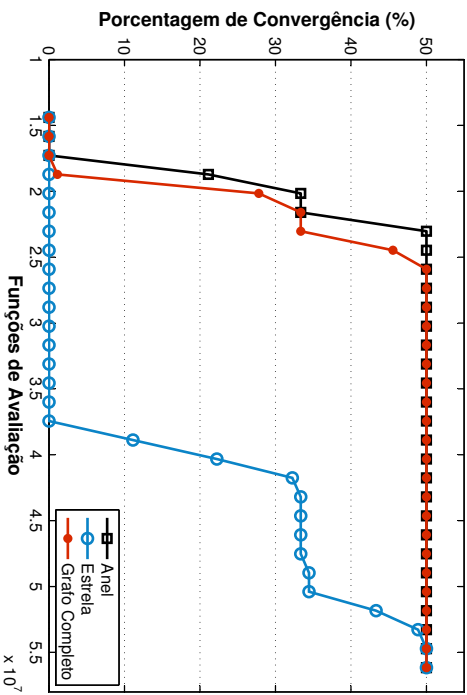
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.4: Taxa de Convergência Acumulativa - Topologia Grafo Completo.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Grafo Completo \times Anel	19969089 \times 19756067	0.8824949
Grafo Completo \times Estrela	19969089 \times 25247133	0.8953287 * 10^{-9}

Tabela 5.10: Comparação entre médias (número de funções de avaliação) em relação à topologia Grafo Completo.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Grafo Completo \times Anel	34.36127 \times 31.52867	0.4098621
Grafo Completo \times Estrela	34.36127 \times 39.83329	0.3155885

Tabela 5.11: Comparação entre médias (melhores soluções) em relação à topologia Grafo Completo.

é composta por vários parâmetros essenciais ao seu desempenho. Neste experimento, todas as combinações de parâmetros apresentados na Tabela 5.12 foram avaliadas. Em sua maioria, as configurações avaliadas não apresentaram diferenças significativas. Assim, os parâmetros usados nos testes comparativos foram $\gamma = 1.0$; $\lambda = 1.0$; $\epsilon = 0.5$; Indicador: Medida de custo média, Reinicialização: nunca, que foi a combinação com melhor desempenho médio estimado.

Parâmetro	Valores
γ	[0.5; 1.0]
λ	[0.5; 1.0]
ϵ	[0.5; 1.0]
Indicador	[Melhores indivíduos; Medida de custo média; Diversidade populacional; Medida de custo média \times Diversidade populacional]
fp (Parâmetro do Indicador Medida de custo média \times Diversidade populacional)	[0.3; 0.5; 0.7]
Reinicialização	[Nunca; a cada 50 gerações; a cada 100 gerações]

Tabela 5.12: Parâmetros avaliados para a topologia Q-learning.

A Figura 5.5 ilustra a distribuição da taxa de convergência acumulada para as diferentes topologias avaliadas. Neste experimento, IM compostos por 4, 8, 16 e 32 ilhas foram testados. Os gráficos demonstram que em modelos com 4 ilhas, a topologia Estrela apresentou a maior taxa final de convergência entre todas topologias. Todavia, as demais topologias necessitam de um número muito inferior de funções de avaliação para se atingir boas soluções do que a topologia Estrela. Para as outras configurações do IM, a topologia Anel é seguida pelas topologias Q-learning e Estrela, respectivamente.

A análise estatística aplicada aos resultados destas topologias indica diferença significativa entre elas. Assim, uma análise post-hoc entre as abordagens comparando o número de funções de avaliação e Δf (qualidade da solução final) foi realizada e seus resultados são apresentados nas Tabelas 5.13 e 5.14, respectivamente.

De acordo com a Tabela 5.13, os resultados indicam superioridade da topologia Q-learning em relação à Estrela e igualdade considerando a topologia Anel. Em relação à media de Δf , vide Tabela 5.14, nenhuma diferença significativa foi detectada.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Q-learning \times Anel	20054289 \times 19756067	0.7916376
Q-learning \times Estrela	20054289 \times 25247133	$0.1625963 * 10^{-8}$

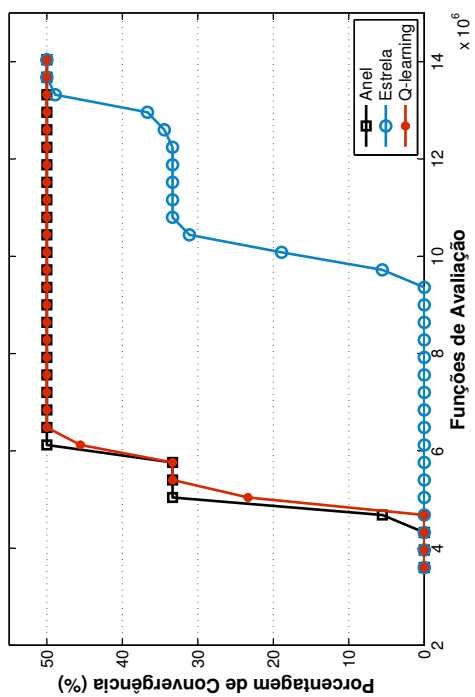
Tabela 5.13: Comparação entre médias (número de funções de avaliação) em relação à topologia Q-learning.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
Q-learning \times Anel	33.82851 \times 31.52867	0.5066778
Q-learning \times Estrela	33.82851 \times 39.83329	0.2469242

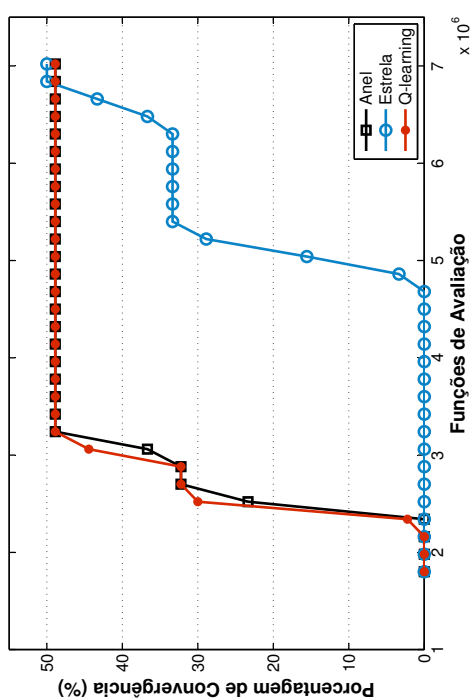
Tabela 5.14: Comparação entre médias (melhores soluções) em relação à topologia Q-learning.

5.2.6 Topologia UCB

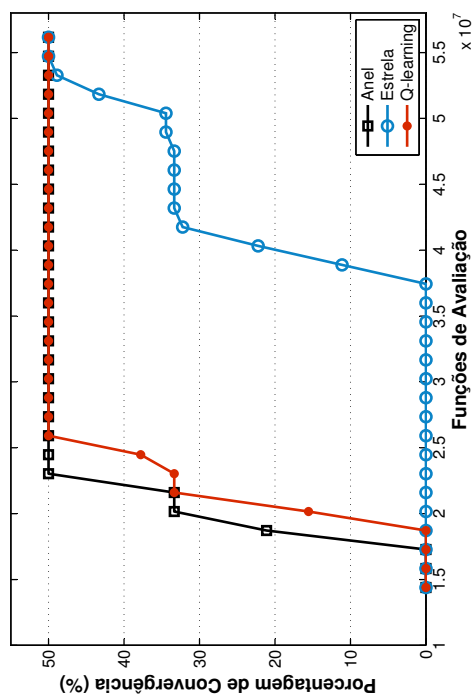
Os resultados apresentados nesta seção são referentes a comparação das topologias Anel, Estrela e UCB. Como estudado no Capítulo 4, a topologia UCB requer a definição



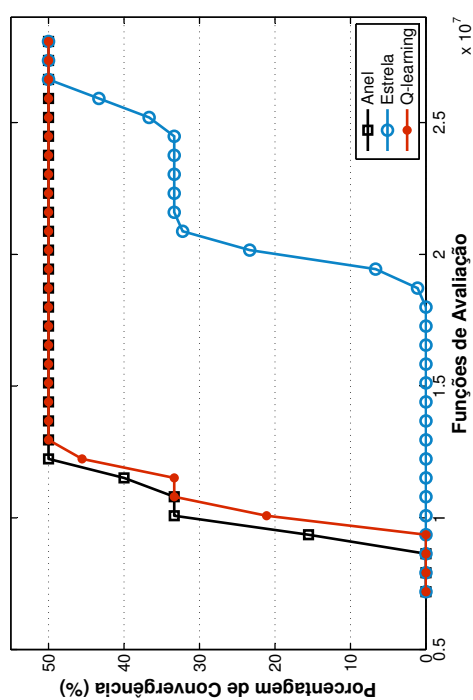
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.5: Convergência Acumulativa - Topologia Q-learning.

de alguns parâmetros de seu mecanismo de configuração. Neste trabalho, diversas configurações foram avaliadas e a Tabela 5.15 apresenta os valores de cada parâmetro testado.

A maioria das configurações avaliadas não apresentaram diferenças significativas. No entanto, a configuração com o melhor desempenho médio foi definida com os seguintes parâmetros: $C = 7.0$; Indicador: Medida de custo média; Reinicialização: a cada 50 gerações. Esta configuração será utilizada durante o restante deste estudo para representar a topologia UCB.

Parâmetro	Valores
C	[5.0; 7.0; 10.0]
Indicador	[Melhores indivíduos; Medida de custo média; Diversidade populacional; Medida de custo média \times Diversidade populacional]
fp (Parâmetro do Indicador Medida de custo média \times Diversidade populacional)	[0.3; 0.5; 0.7]
Reinicialização	[Nunca; a cada 50 gerações; a cada 100 gerações]

Tabela 5.15: Parâmetros avaliados para a topologia UCB.

A Figura 5.6 descreve a distribuição da taxa de convergência para as diferentes topologias avaliadas utilizando modelos com 4, 8, 16 e 32 ilhas. Os gráficos demonstram que a topologia UCB foi competitiva com a topologia Anel somente para modelos com 4 e 8 ilhas. Em todas as configurações do IM esta topologia dinâmica estudada foi melhor que a topologia Estrela. Além do mais, a topologia Estrela obteve o pior desempenho entre as topologias de migração comparadas.

A análise de variância aplicada aos resultados do IM empregando as topologias Anel, Estrela e UCB sugere a refutação da hipótese de igualdade entre suas médias. Em seguida, uma análise comparativa entre os resultados das topologias foi realizada e resumida nas Tabelas 5.16 e 5.17.

Em relação ao número de funções de avaliação, a topologia UCB foi significamente melhor que a Estrela, por outro lado, a mesma não apresentou diferença em relação à topologia Anel, para um nível de significância de 95%. Considerando a média da qualidade das soluções encontradas, os resultados da análise post-hoc indicam que as

diferenças detectadas não são significativas.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
UCB \times Anel	20185867 \times 19756067	0.7063704
UCB \times Estrela	20185867 \times 25247133	0.3156724 * 10^{-8}

Tabela 5.16: Comparação entre médias (número de funções de avaliação) em relação à topologia UCB.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
UCB \times Anel	33.11761 \times 31.52867	0.6454387
UCB \times Estrela	33.11761 \times 39.83329	0.1686034

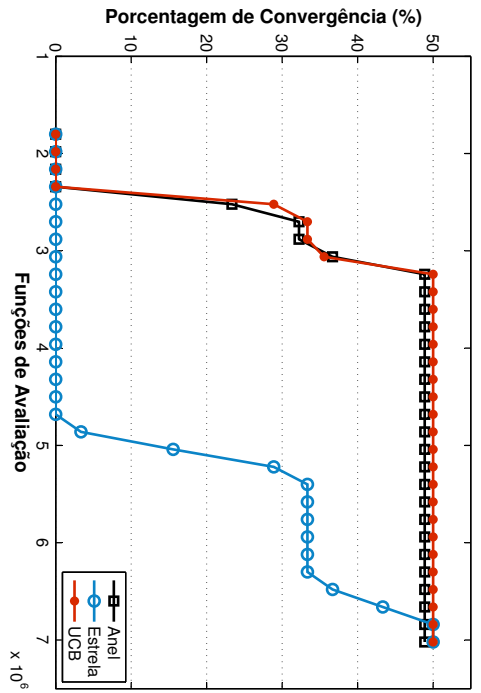
Tabela 5.17: Comparação entre médias (melhores soluções) em relação à topologia UCB.

5.2.7 Topologia K-medoids

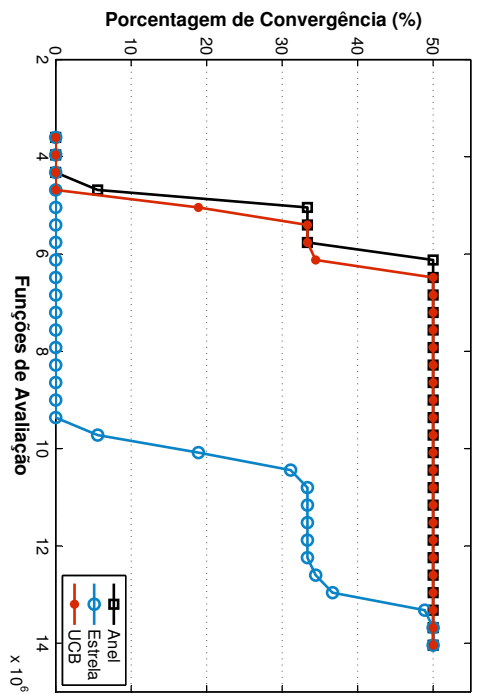
Esta seção descreve os resultados comparativos entre a topologia dinâmica K-medoids e as topologias estáticas Anel e Estrela. O mecanismo de configuração que compõe esta topologia dinâmica exige a definição do número de grupos, que se deseja dividir os dados, além de uma métrica de distância entre soluções. Neste experimento, as métricas de distância *Euclidiana* e *Manhattan*, descritas na seção 4.4.2 foram consideradas, bem como as topologias Aleatória e Anel foram avaliadas para construção das topologias intra-grupos.

Testes estatísticos indicaram que não existem diferenças significativas entre as configurações da topologia dinâmica K-medoids. Desta maneira, a configuração com a menor média em relação ao número de funções de avaliação foi escolhida. Para esta configuração escolhida, os parâmetros de seu mecanismo de configuração são: Número de grupos: 2; Métrica de distância: Euclidiana; Topologia intra-grupo: Anel.

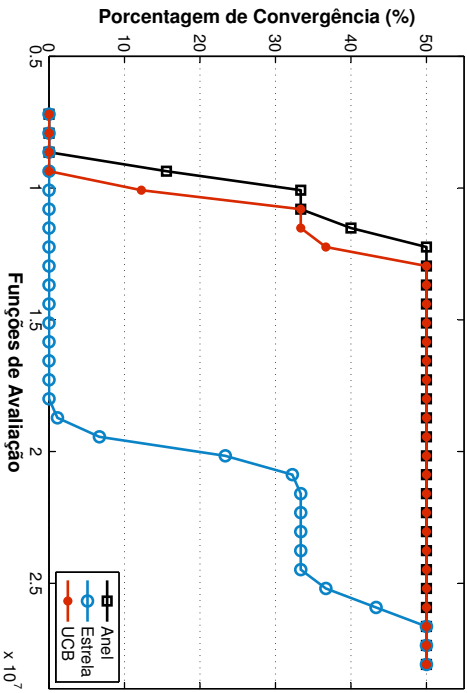
Na Figura 5.7, a distribuição da taxa de convergência das topologias avaliadas é apresentada para modelo com 4, 8, 16 e 32 ilhas. Em todas as configurações, a topologia



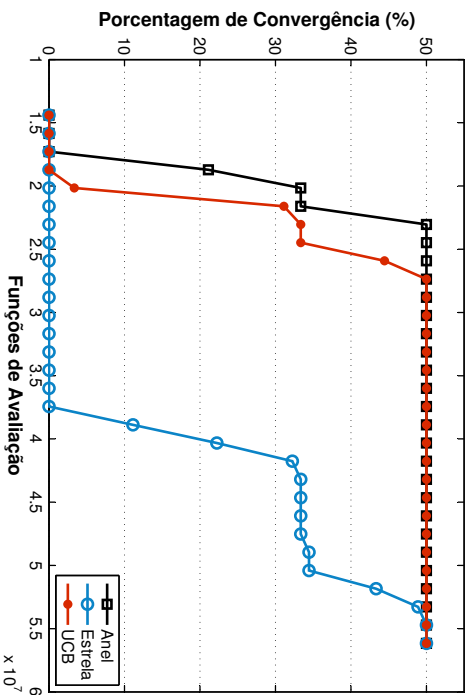
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.6: Convergência Acumulativa - Topologia UCB.

K-medoids foi superada pela topologia Anel. Entretanto, ela foi superior a topologia Estrela, que obteve o pior desempenho entre as topologias avaliadas.

Como análise de variância das médias indicou diferença significativa entre as médias, a análise post-hoc foi realizada e seus resultados sobre as variáveis de resposta (número de funções de avaliação e qualidade da solução final) são apresentados nas Tabelas 5.18 e 5.19. Estes resultados apresentam as respectivas médias dessas variáveis e os valores-p ajustados para cada comparação aplicada.

Analisando os valores-p encontrados, é possível inferir que somente a topologia K-medoids foi superior à Estrela no quesito número de funções de avaliação, dado 95% de confiança. Os demais testes sugerem que não existem diferenças significativas entre as topologias.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
K-medoids \times Anel	20064444 \times 19756067	0.6810715
K-medoids \times Estrela	20064444 \times 25247133	$0.2789105 * 10^{-8}$

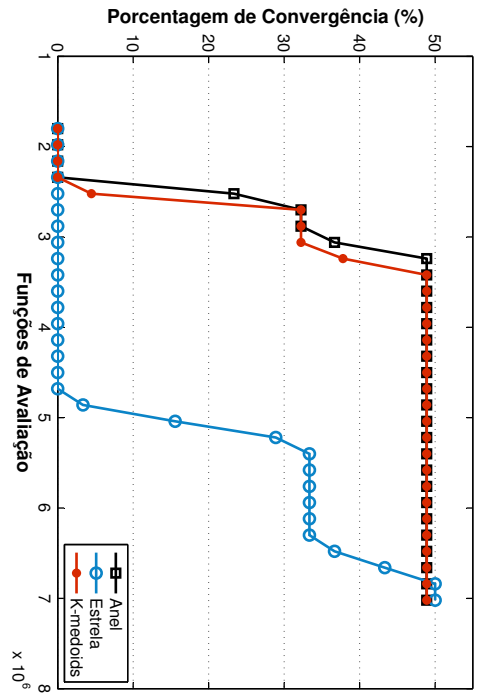
Tabela 5.18: Comparação entre médias (número de funções de avaliação) em relação à topologia K-medoids.

Topologias Avaliadas	Médias Avaliadas	Valor p ajustado
K-medoids \times Anel	32.31573 \times 31.52867	0.8170785
K-medoids \times Estrela	32.31573 \times 39.83329	0.1013314

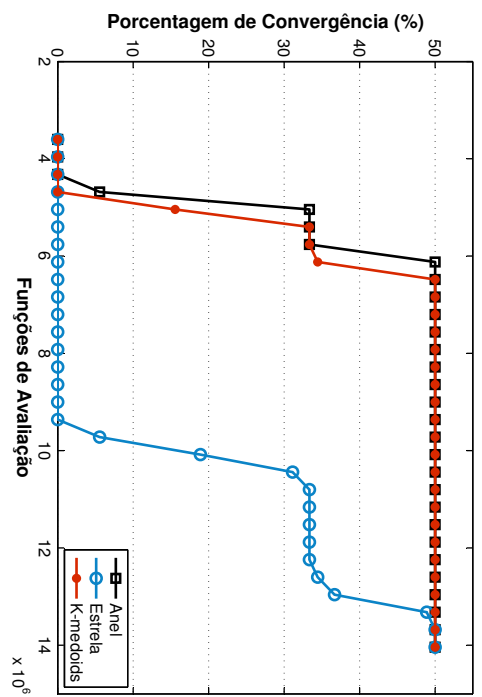
Tabela 5.19: Comparação entre médias (melhores soluções) em relação à topologia K-medoids.

5.2.8 Resultados Comparativos: Topologias Dinâmicas

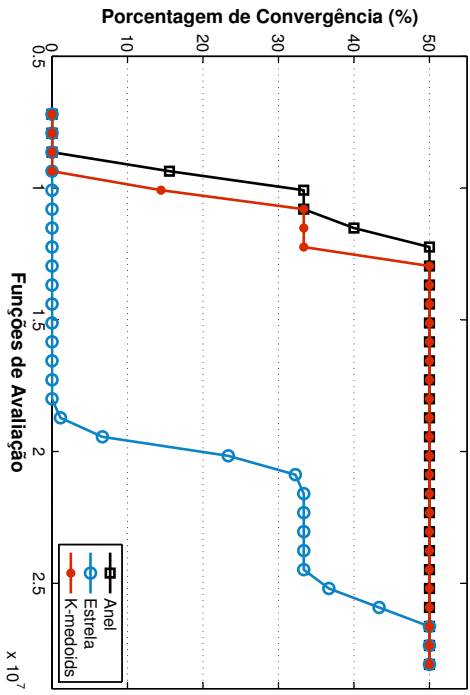
Esta seção apresenta um comparativo entre todas as topologias dinâmicas estudadas neste trabalho. As Figuras 5.8, 5.9, 5.10 e 5.11 descrevem a distribuição da taxa de convergência acumulada do tempo de execução para todas as topologias dinâmicas. Nestes testes, IM compostos por 4, 8, 16 e 32 ilhas foram avaliados e os resultados



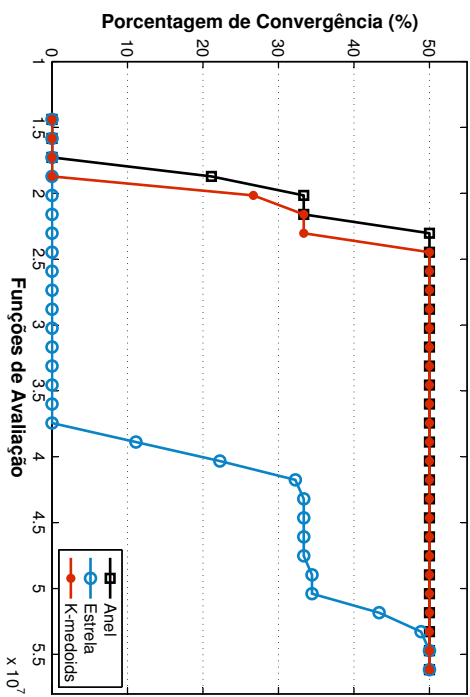
(a) 4 ilhas



(b) 8 ilhas



(c) 16 ilhas



(d) 32 ilhas

Figura 5.7: Convergência Acumulativa - Topologia K-medoids.

sugerem uma vantagem das topologias Roleta, Torneio e Grafo Completo, em relação às demais.

Nota-se que, entre as melhores topologias dinâmicas, a topologia Roleta apresentou o melhor desempenho, seguido pelas topologias Torneio e Grafo Completo. O restante das topologias dinâmicas apresentaram desempenhos similares, sendo que, as topologias Q-learning e K-medoids foram os maiores destaques. A topologia Q-learning foi ligeiramente superior às topologias Aleatória, K-medoids e UCB em modelos com 8 e 16 ilhas. A topologia K-medoids destacou-se devido a melhora em seu desempenho com o aumento do número de ilhas do modelo.

Testes estatísticos considerando o número de funções de avaliação e a qualidade da solução final foram realizados. Assim, com 95% de confiança, os testes estatísticos comparando as médias do número de funções de avaliação indicam que não existem diferenças significativas entre os resultados das topologias dinâmicas. Da mesma forma, a análise estatística considerando a qualidade da solução final não detectou diferença entre as topologias estudadas.

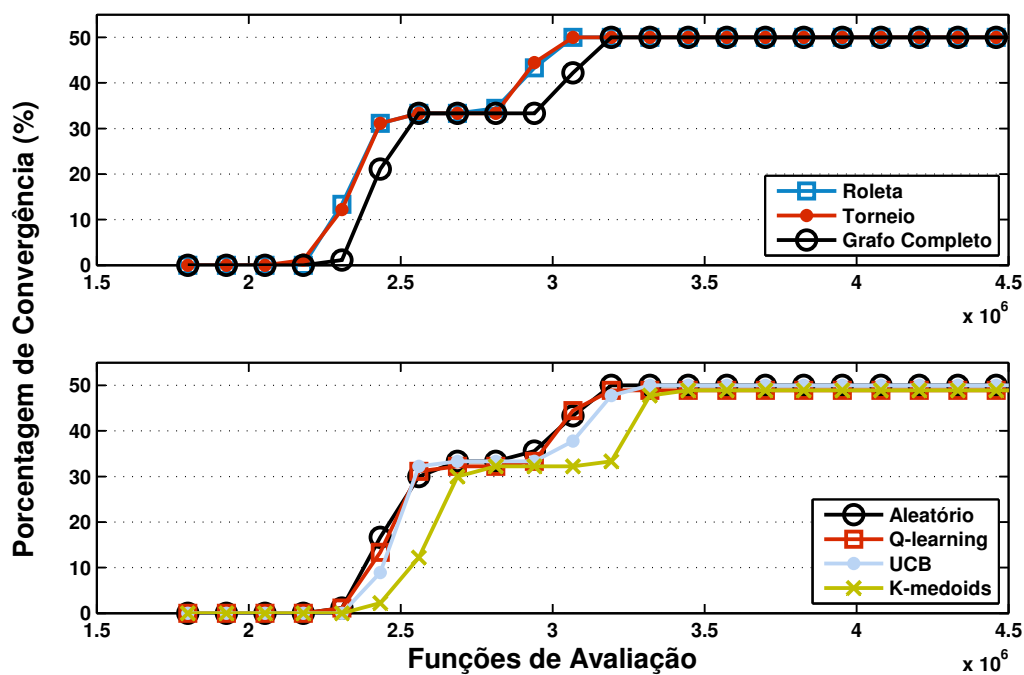


Figura 5.8: Convergência Acumulativa - Topologias Dinâmicas - 4 Ilhas

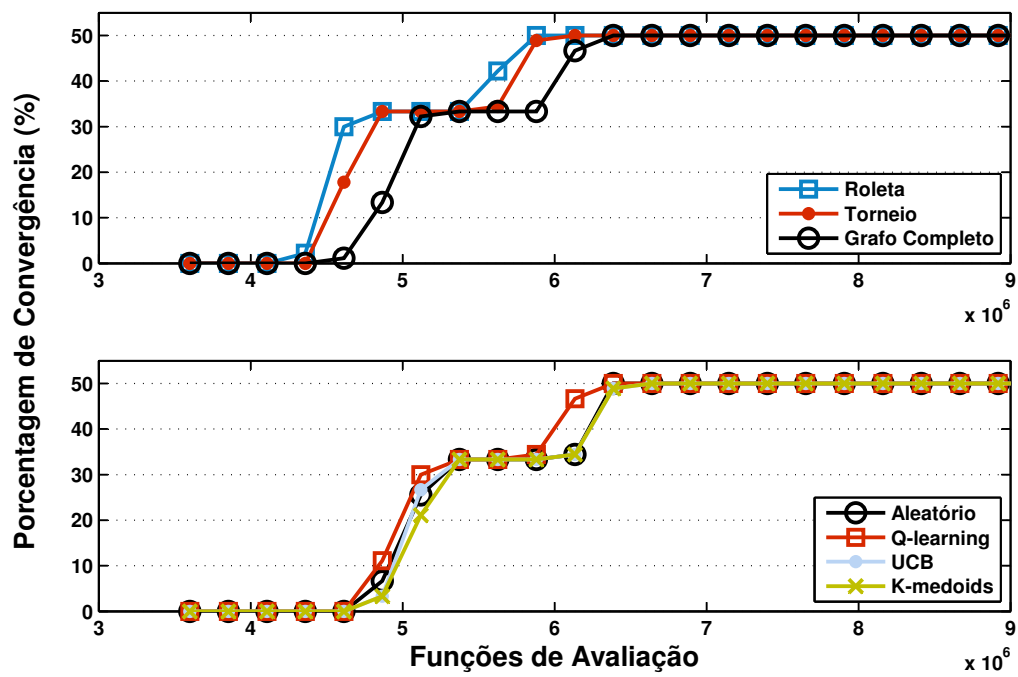


Figura 5.9: Convergência Acumulativa - Topologias Dinâmicas - 8 Ilhas

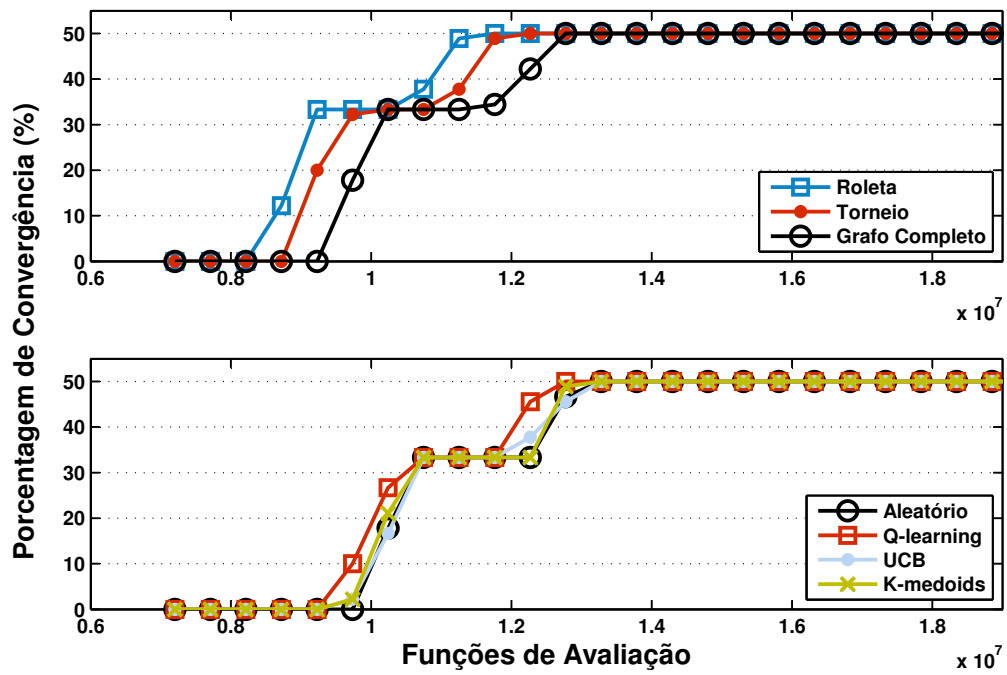


Figura 5.10: Convergência Acumulativa - Topologias Dinâmicas - 16 Ilhas

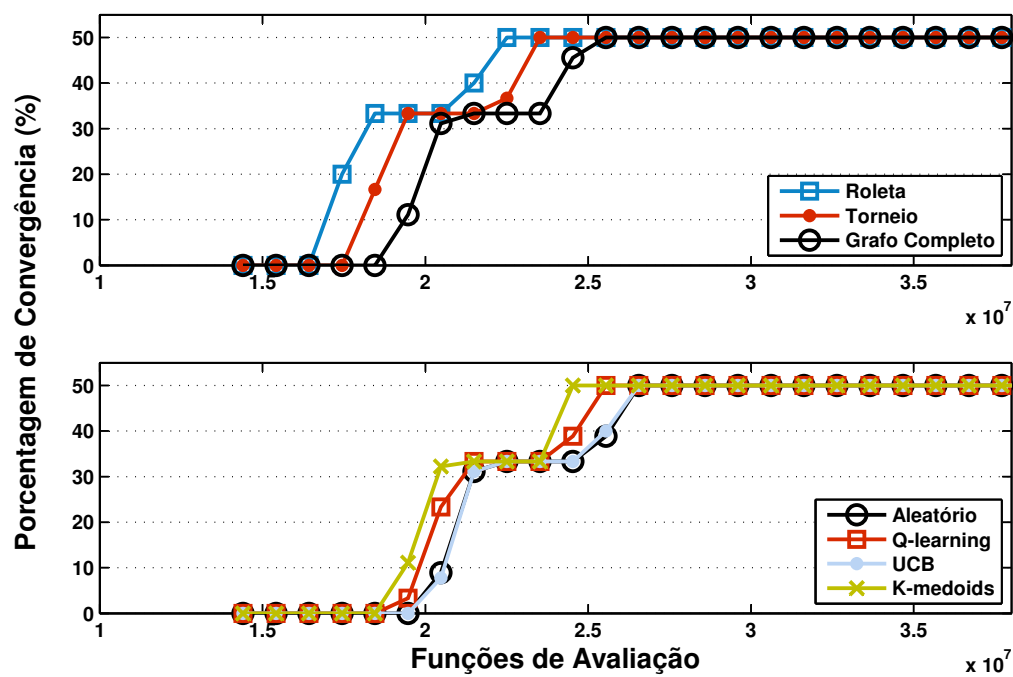


Figura 5.11: Convergência Acumulativa - Topologias Dinâmicas - 32 Ilhas

5.3 Comportamento das Topologias

Nesta seção, os fluxos migratórios gerados pelas topologias de migração são visualizados a partir do ponto de vista de redes complexas (Barabási 2003), (Viana 2007). Redes complexas descrevem grafos que apresentam estruturas não-triviais. Seu formalismo é muito útil para descrever problemas que possuem grande número de agentes que se interagem (Viana 2007). Os grafos utilizados para descrever as redes complexas são representados por um conjunto de vértices (nós) e um conjunto de conexões. As conexões são conhecidas como arestas e estabelecem relações entre os vértices. As arestas podem apresentar orientação quanto a sua direção ou não.

Este estudo visa compreender melhor o comportamento dos fluxos migratórios durante o processo evolutivo do IM através de redes complexas. Para isto, três métricas foram escolhidas para quantificar diferentes características dos grafos. A primeira métrica definida foi o *diâmetro* (Boccaletti, Latora, Moreno, Chavez & Hwang 2006), na qual mensura o maior caminho do grafo. A segunda foi o *caminho mínimo médio* (Boccaletti, Latora, Moreno, Chavez & Hwang 2006), medida que quantifica a média dos caminhos mínimos do grafo. Por fim, o *maior grau de conexões de saída* (Boccaletti, Latora, Moreno, Chavez & Hwang 2006) também foi escolhido, o qual mede a quantidade máxima de arestas saindo entre todos os nós.

Cada métrica apresentada representa significados distintos em nosso contexto de topologias de migração para IM. O diâmetro indica o quão compacta a topologia de migração é e fornece a ideia do maior atraso de propagação da informação. O caminho médio revela a ideia de atraso médio de propagação de uma boa solução para todas as ilhas. O maior grau de conexões de saída sugere a informação de concentração de conexões em algumas subpopulações do modelo.

Devido ao contraste de seus desempenhos e os valores das métricas de redes complexas, todas as topologias dinâmicas e estáticas foram incluídas neste experimento. Assim, este experimento tem a finalidade de fornecer uma ideia de como a topologia de migração deve ser.

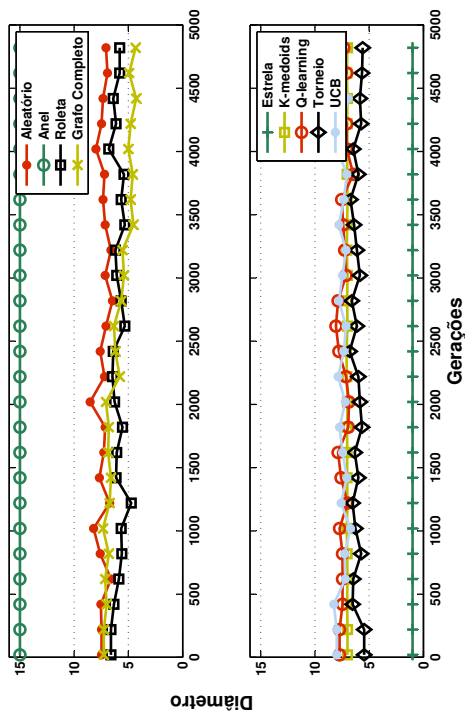
A Figura 5.12 representa o diâmetro médio das topologias estudadas em todas as execuções com 16 e 32 ilhas. Nestes experimentos, as funções de teste f_1 (unimodal) e f_5 (multimodal) foram consideradas. Note que a topologia Anel apresenta os maiores valores de diâmetro, com a Estrela apresentando o mais baixo. As topologias dinâmicas apresentaram valores intermediários, com uma tendência para a extremidade inferior.

Combinando estes experimentos e o desempenho das abordagens, os resultados indicam que estruturas compactas em demasia não são interessantes em termos de desempenho do IM se não há mudanças dinâmicas dos fluxos. No entanto, topologias com altos valores de diâmetro, como a topologia Anel, também não são necessárias.

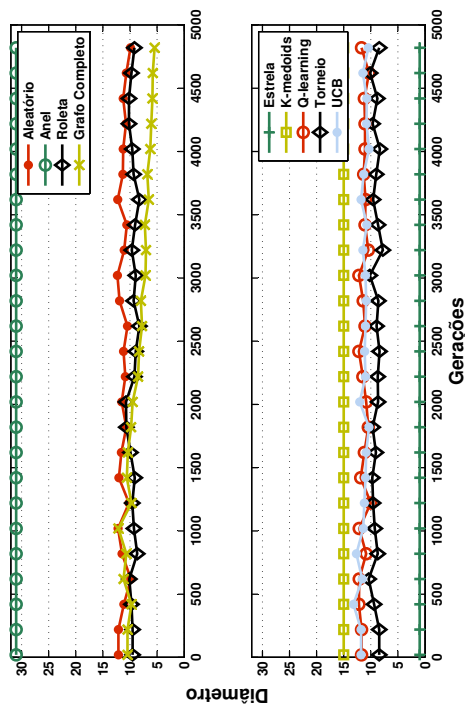
Todas as abordagens dinâmicas e a topologia Estrela apresentaram valores menores de diâmetro, sendo a Estrela o menor. Por outro lado, a topologia Estrela não é capaz de identificar a ilha de melhor desempenho em cada etapa do processo evolutivo, pois é definida a priori. Enquanto seu diâmetro é o menor de todas as topologias avaliadas, esta abordagem não garante que boas soluções sejam propagadas.

Na Figura 5.13 o caminho médio das topologias é apresentado em todas as execuções para modelos com 16 e 32 ilhas. As funções de teste avaliadas neste experimento são: f_1 e f_5 . Novamente, as abordagens dinâmicas apresentaram valores intermediários e a Estrela o menor. A topologia Anel registrou o maior caminho médio entre todas. Assim, as mesmas observações feitas a respeito do comparativo do desempenho das abordagens estudadas e o diâmetro topologias de migração podem ser consideradas para esta métrica.

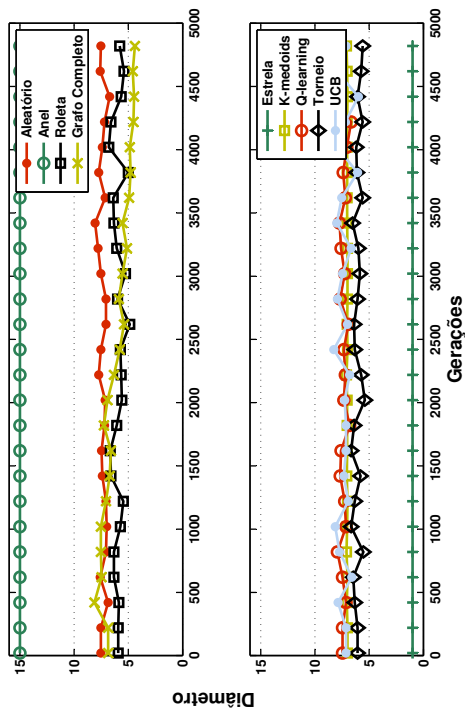
A Figura 5.14 representa a média do número máximo de conexões de saída dos grafos gerados em todas as execuções do IM. Estes resultados referem-se a IM com 16 e 32 ilhas aplicados as funções de teste f_1 e f_5 . Mais uma vez, quando vista em contraste com os experimentos de desempenho, estes resultados indicam que é interessante para um IM manter um certo nível de conexões em uma subpopulação, que provavelmente estimula a diversidade e a cooperação entre as ilhas do modelo.



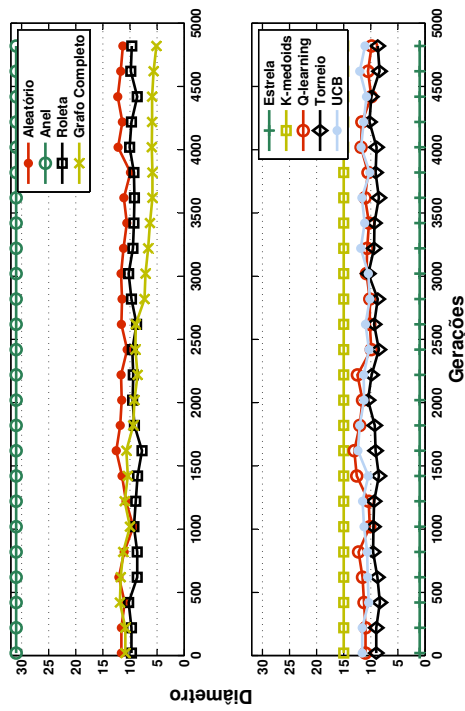
(b) Função f_5 - 16 ilhas



(d) Função f_5 - 32 ilhas

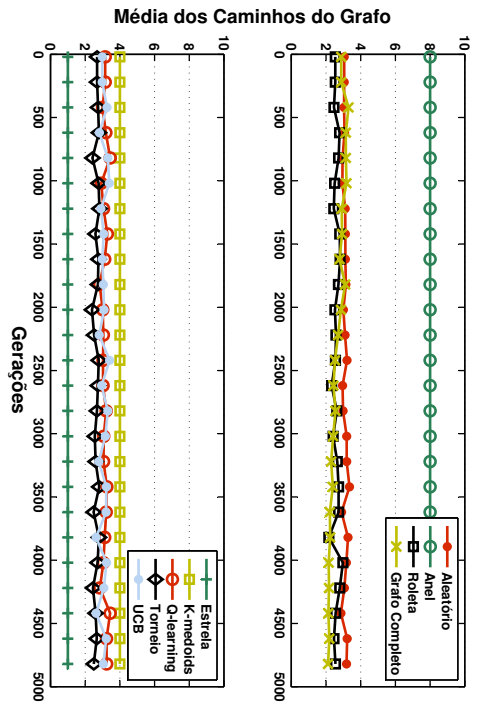


(a) Função f_1 - 16 ilhas

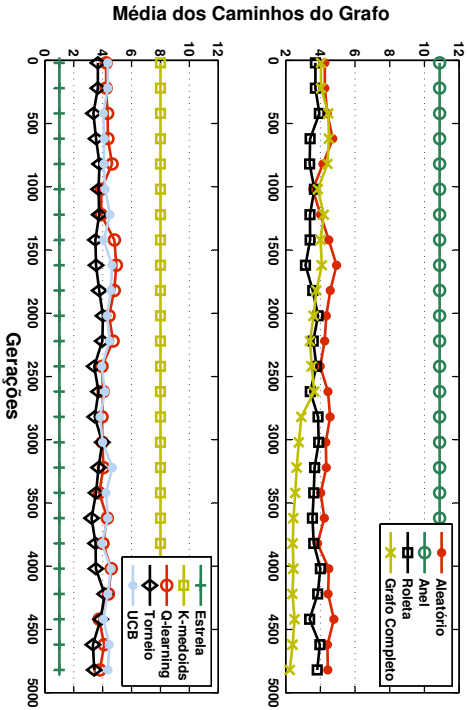


(c) Função f_1 - 32 ilhas

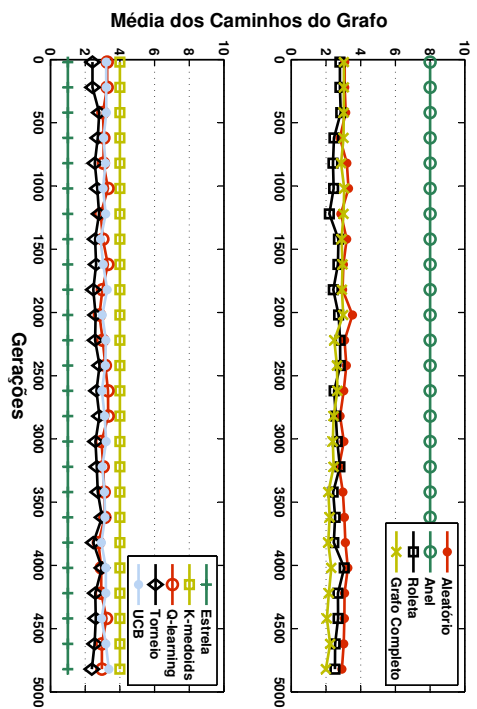
Figura 5.12: Diâmetro Médio dos Grafos.



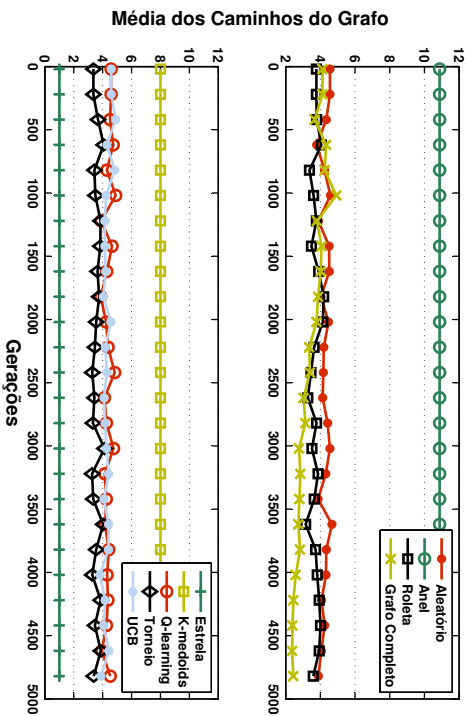
(a) Função f_1 - 16 ilhas



(c) Função f_1 - 32 ilhas

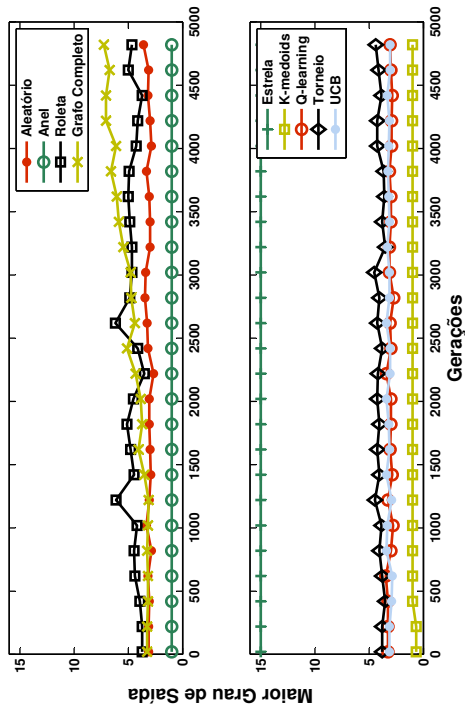


(b) Função f_5 - 16 ilhas

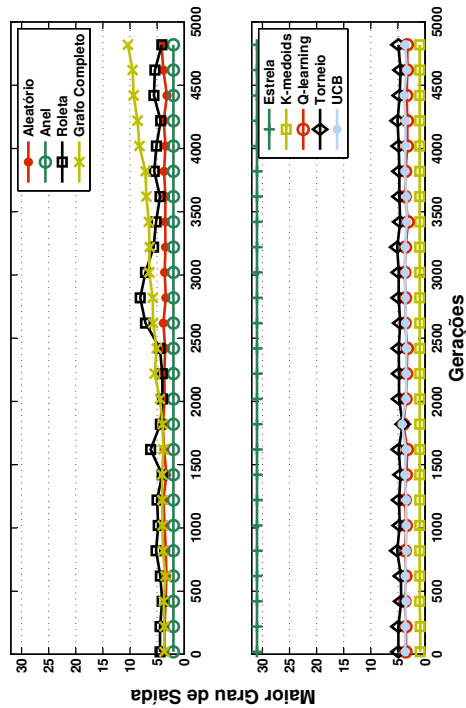


(d) Função f_5 - 32 ilhas

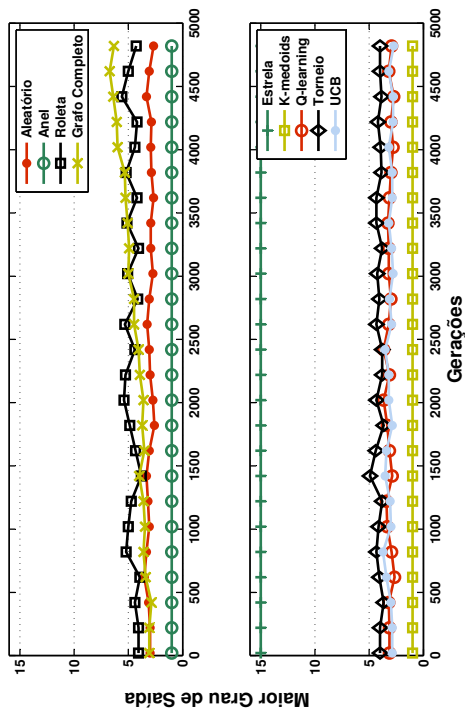
Figura 5.13: Média dos Caminhos Mínimos dos Grafos.



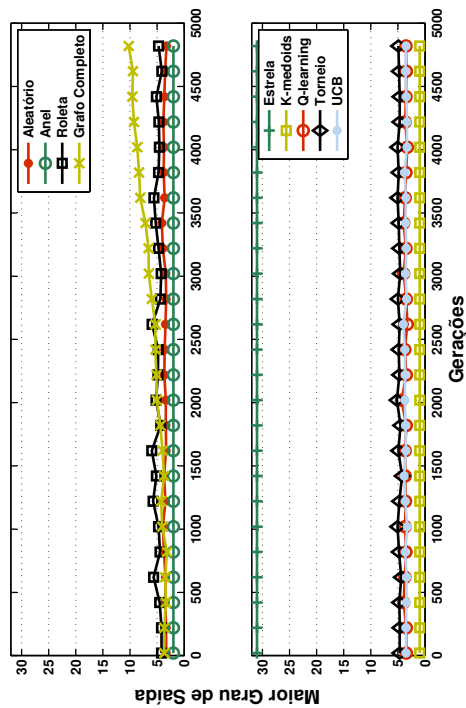
(b) Função f_5 - 16 ilhas



(d) Função f_5 - 32 ilhas



(a) Função f_1 - 16 ilhas



(c) Função f_1 - 32 ilhas

Figura 5.14: Média do Maior Grau de Conexões de Saída.

Capítulo 6

Conclusões

Nos últimos anos, as estratégias de paralelização de metaheurísticas populacionais são promissoras formas de manter estes métodos competitivos diante de problemas de otimização complexos. Estas estratégias economizam tempo distribuindo o esforço computacional e podem se beneficiar de uma configuração paralela a partir do ponto de vista algorítmico. Em particular, a estratégia Modelo em Ilhas, proposta na década de 1980, vem se destacando devido ao seu desempenho e quantidade de metaheurísticas em que tem sido aplicado com enorme sucesso. Ela tem se mostrado popular e eficiente tanto em arquiteturas computacionais paralelas como em sequenciais.

No IM, apesar do relativo isolamento das ilhas, o processo de migração é uma fundamental característica para se alcançar o objetivo em um processo evolutivo. Os fluxos direcionados criados pelo processo de migração formam uma estrutura de vizinhança conhecida como topologia de migração. Esta estrutura representa uma importante regra para o IM. Diversos estudos encontrados na literatura indicam que elas exercem efetiva influência no desempenho dos modelos.

Em sua maioria as topologias são ingênuas e baseadas em redes e arquiteturas de computadores não levando em consideração a dinâmica do processo evolutivo. Neste contexto, diversas abordagens tem sido propostas na literatura apresentando topologias dinâmicas capazes de se reconfigurar durante o processo de otimização. Com o objetivo de estudar as topologias dinâmicas em IM, neste trabalho, foi definido um modelo utilizando o algoritmo de Evolução Diferencial para uma avaliação de desempenho e comportamento dos fluxos gerados por elas.

De modo geral, os resultados obtidos sugerem que as topologias dinâmicas podem

ser superiores às topologias estáticas clássicas, alcançando boas soluções em um número muito inferior de funções de avaliação. Outro resultado geral importante é que elas são capazes de manter a qualidade das soluções finais encontradas, mesmo diminuindo o número de avaliações da função objetivo.

Em relação as topologias estáticas, todas as abordagens dinâmicas estudadas foram superiores à topologia Estrela e pelo menos competitivas com a Anel. Os maiores destaques comparados à topologia Anel foram as topologias Roleta e Torneio. A média do número de funções de avaliação considerando todos os experimentos alcançados por elas foi menor que a topologia estática Anel. Em IM com 4, 8 e 16 ilhas este comportamento é claramente evidenciado. Além disto, este desempenho foi possível sem que houvesse perda significativa na qualidade das soluções.

Entre as abordagens dinâmicas, mesmo não apresentando diferenças significativas entre si, os destaques ficaram por conta das topologias Grafo Completo, Roleta e Torneio. A topologia Roleta apresentou o melhor desempenho entre elas para todas as configurações do IM avaliadas. A topologia Torneio consagrou-se a segunda melhor sendo seguida pela topologia Grafo Completo. Outra vantagem destas topologias, em relação as demais dinâmicas, é a complexidade de seus mecanismos de configuração, que são facilmente implementados e paralelizados.

Os resultados das demais topologias dinâmicas indicam que elas estão abaixo do esperado sendo competitivas com a melhor topologia estática, apenas em IM menores, como por exemplo, modelos com 4 e 8 subpopulações. Além do mais, a maioria destas topologias apresentam mecanismos difíceis de serem implementados e ajustados.

O estudo do comportamento das topologias geradas pelos mecanismos de configuração durante o processo evolutivo demonstrou que a maioria das abordagens apresentam valores intermediários para as métricas de redes complexas. As informações das métricas estudadas sugerem que as topologias definidas não são compactas em demasia e tão pouco extensas. Estas características indicam que as topologias dinâmicas apresentam boa capacidade de propagação de informações pelas ilhas.

A conclusão geral deste estudo é que as topologias dinâmicas são promissoras formas de se configurar o processo de migração no Modelo em Ilhas. Elas são capazes de gerar estruturas mais espertas que as tradicionais. Mesmo mecanismos de configuração simples são capazes de compreender a dinâmica do processo evolutivo.

No entanto, essas considerações se limitam aos experimentos realizados. Outros parâ-

metros do Modelo em Ilhas e do método de otimização devem ser considerados devido à influência que eles exercem sobre o modelo. Como trabalhos futuros, o autor considera a variação destes parâmetros com o intuito de melhor inferir o desempenho das topologias de migração estáticas e dinâmicas. Uma análise sobre o tempo computacional também é recomendada pois as topologias estudadas apresentam diferenças na complexidade de seus mecanismos de configuração.

Em relação ao comportamento das topologias de migração, o estudo realizado possibilita a indução de novas topologias baseadas em fundamentos de Redes Complexas. A inferência de topologias como redes de pequeno mundo pode indicar novos rumos em relação às topologias de migração em Modelo em Ilhas.

Apêndice A

Apêndice

A.1 Funções de Teste

A.1.1 Função 1: Sphere Deslocada

$$f_1(x) = \sum_{i=1}^n z_i^2 + f_bias_1, z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.1})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-100, 100]$
- Ótimo global: $x^* = o$
- $f_1(x^*) = f_bias_1 = -450$

A.1.2 Função 2: Schwefel Deslocada

$$f_2(x) = \max \{ \|z_i\|, 1 \leq i \leq n \} + f_bias_2, z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.2})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-100, 100]$
- Ótimo global: $x^* = o$
- $f_2(x^*) = f_bias_2 = -450$

A.1.3 Função 3: Rosenbrock Deslocada

$$f_3(x) = \sum_{i=1}^{n-1} (100(z_i^2 - z_{i-1})^2 + (z_i - 1)^2) + f_bias_3, z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.3})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-100, 100]$
- Ótimo global: $x^* = o$
- $f_3(x^*) = f_bias_3 = 390$

A.1.4 Função 4: Rastrigin Deslocada

$$f_4(x) = \sum_{i=1}^n (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_bias_4, z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.4})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-5, 5]$
- Ótimo global: $x^* = o$
- $f_4(x^*) = f_bias_4 = -330$

A.1.5 Função 5: Griewank Deslocada

$$f_5(x) = \sum_{i=1}^n \frac{z_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_bias_5, z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.5})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-600, 600]$
- Ótimo global: $x^* = o$
- $f_5(x^*) = f_bias_5 = -180$

A.1.6 Função 6: Ackley Deslocada

$$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi z_i)\right) + 20 + e + f_bias_6, \\ z = x - o, x = [x_1, x_2, \dots, x_n] \quad (\text{A.6})$$

onde n é o número de dimensões do problema e $o = [o_1, o_2, \dots, o_n]$ é o ótimo global deslocado.

Características:

- $x \in [-32, 32]$
- Ótimo global: $x^* = o$

- $f_6(x^*) = f_bias_6 = -140$

Referências Bibliográficas

- Adamidis, P. (1998). Parallel evolutionary algorithms: A review, *4th Hellenic-European Conference on Computer Mathematics and its Applications*.
- Alba, E. (2002). Parallel evolutionary algorithms can achieve super-linear performance, *Information Processing Letters* **82**: 7–13.
- Alba, E. & Tomassini, M. (2002). Parallelism and evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* **6**: 443–462.
- Alba, E. & Troya, J. M. (1999). A survey of parallel distributed genetic algorithms, *Complexity* **4**: 31–52.
- Auer, P. (2003). Using confidence bounds for exploitation-exploration trade-offs, *Journal of Machine Learning Research* **3**: 397–422.
- Auer, P., Cesa-Bianchi, N. & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem, *Machine Learning* **47**(2–3): 235–256.
- Bäck, T., Fogel, D. B. & Michalewicz, Z. (2000a). *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol.
- Bäck, T., Fogel, D. B. & Michalewicz, Z. (2000b). *Evolutionary Computation 2: Advanced Algorithms and Operators*, Institute of Physics Publishing, Bristol.
- Bäck, T., Hammel, U. & Schwefel, H.-P. (1997). Evolutionary computation: comments on the history and current state, *IEEE Transactions on Evolutionary Computation* pp. 3–17.
- Barabási, A. (2003). *Linked: how everything is connected to everything else and what it means for business, science, and everyday life*, Plume book, Plume.

- Bazaraa, M., Sherali, H. & Shetty, C. (2006). *Nonlinear Programming: Theory and Algorithms*, 3 edn, John Wiley & Sons.
- Belding, T. C. (1995). The Distributed Genetic Algorithm Revisited, in L. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA, pp. 114–121.
- Benjamini, Y. & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *Journal of the Royal Statistical Society. Series B (Methodological)* **57**: 289–300.
- Berntsson, J. & Tang, M. (2005). Dynamic optimization of migration topology in internet-based distributed genetic algorithms, *Proceedings of the 2005 conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, pp. 1579–1580.
- Berntsson, L. J. (2006). *An adaptive framework for Internet-based distributed genetic algorithms*, PhD thesis, Queensland University of Technology.
- Bethke, A. (1976). *Comparison of genetic algorithms and gradient-based optimizers on parallel processors: efficiency of use of processing capacity*, Technical report, University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department.
- Blickle, T. (1996). *Theory of Evolutionary Algorithms and Application to System Synthesis*, PhD thesis, Swiss Federal Institute of Technology, Zurich.
- Blum, C. & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys*, Vol. 35, ACM, New York, NY, USA, pp. 268–308.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. (2006). Complex networks: Structure and dynamics, *Physics Reports, Elsevier* **424**: 175–308.
- Cantú-Paz, E. (1998). A survey of parallel genetic algorithms, *Calculateurs Paralleles, Reseaux Et Systems Repartis* **10**: 141–171.
- Cantú-Paz, E. (2001). Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms, *Journal Of Heuristics* **7**: 311–334.
- Cantú-Paz, E. (n.d.). Designing efficient master-slave parallel genetic algorithms, *ILLI-GAL Technical Report No. 97004* .

- Cantú-Paz, E. & Goldberg, D. E. (1999). On the scalability of parallel genetic algorithms, *Evolutionary Computation* pp. 429–449.
- Chakraborty, U. K. (2008). *Advances in Differential Evolution*, 1 edn, Springer Publishing Company, Incorporated.
- Cochran, W. G., Cochran, W. & Cox, G. M. (1992). *Experimental designs*, Wiley Classics Library, Wiley.
- Cohon, J. P., Hegde, S. U., Martin, W. N. & Richards, D. (1987). Punctuated equilibria: a parallel genetic algorithm, *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp. 148–154.
- Darwin, C. (1872). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, sixth edn, J. Murray.
- Das, S. & Suganthan, P. (2011). Differential evolution: A survey of the state-of-the-art, *Evolutionary Computation, IEEE Transactions on* **15**(1): 4–31.
- de Castro, L. N. (2007). Fundamentals of natural computing: an overview, *Physics of Life Reviews* **4**(1): 1–36.
- Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L. & White, A. (eds) (2003). *Sourcebook of parallel computing*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Fogel, D. B. (1994). Asymptotic Convergence Properties of Genetic Algorithms and Evolutionary Programming: Analysis and Experiments, *Cybernetics and Systems*, pp. 389–407.
- Fogel, D. B. (1997). The advantages of evolutionary computation, *Biocomputing and emergent computation: Proceedings of BCEC97*, World Scientific Press, pp. 1–11.
- Fogel, L., Owens, A. & Walsh, M. (1966). *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, Chichester, WS, UK.
- Garcia-Baleon, H. A., Alarcon-Aquino, V. & Starostenko, O. (2009). K-medoids-based random biometric pattern for cryptographic key generation, *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Springer-Verlag, Berlin, Heidelberg, pp. 85–94.

- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Goldberg, D. E. & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms, in G. J. E. Rawlins (ed.), *Foundations of Genetic Algorithms*, San Francisco, CA: Morgan Kaufmann, pp. 69–93.
- Golub, M. & Budin, L. (2000). An asynchronous model of global parallel genetic algorithms, *Second ICSC Symposium on Engineering of Intelligent Systems EIS2000*, ICSC Academic Press, Paisley, Scotland, UK, pp. 353–359.
- Gorges-Schleuter, M. (1989). Asparagos an asynchronous parallel genetic optimization strategy, *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 422–427.
- Gorges-Schleuter, M. (1990). Explicit parallelism of genetic algorithms through population structures, *PPSN*, pp. 150–159.
- Guimarães, F. G. (2009). Algoritmos de evolução diferencial para otimização e aprendizado de máquina, *Anais do IX Congresso Brasileiro de Redes Neurais*, Vol. 1, Rio de Janeiro, RJ, Brasil, pp. 1–17.
- Gulsen, M. & Smith, A. E. (1999). *A hierarchical genetic algorithm for system identification and curve fitting with a supercomputer implementation*, Master's thesis, University of Pittsburgh, Department of Industrial Engineering.
- Hinterding, R., Michalewicz, Z. & Eiben, A. E. (1997). Adaptation in evolutionary computation: A survey, *In Proceedings of the Fourth International Conference on Evolutionary Computation ICEC 97*, IEEE Press, pp. 65–69.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA.
- Jaddan, O. A., Rajamani, L. & Rao, C. R. (2005). Improved selection operator for ga, *Journal of Theoretical and Applied Information Technology* pp. 269–277.
- Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996). Reinforcement learning: a survey, *Journal of Artificial Intelligence Research* 4: 237–285.
- Karaboga, N. (2005). Digital iir filter design using differential evolution algorithm, *EURASIP Journal on Advances in Signal Processing*, Hindawi Publishing Corp., New York, NY, United States, pp. 1269–1276.

- Kaufman, L. & Rousseeuw, P. (1987). *Clustering by Means of Medoids*, Vol. 3 of *Reports of the Faculty of Mathematics and Informatics. Delft University of Technology, Fac., Univ.*
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, *Neural Networks, 1995. Proceedings., IEEE International Conference on*, IEEE, pp. 1942–1948.
- Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, USA.
- Krishnapuram, R., Joshi, A. & Yi, L. (1999). A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering, *Snippet Clustering, in Proc. IEEE Intl. Conf. Fuzzy Systems - FUZZIEEE99, Korea*, Vol. 3, pp. 1281–1286.
- Lardeux, F. & Goëffon, A. (2010). A dynamic island-based genetic algorithms framework, *Proceedings of the 8th international conference on Simulated evolution and learning*, Springer-Verlag, Berlin, Heidelberg, pp. 156–165.
- Lima, C. R. E., Lopes, H. S. & Tavares, L. G. (2009). A study of topology in insular parallel genetic algorithms., *Nature & Biologically Inspired Computing*, IEEE, pp. 632–635.
- Lopes, R. A., Silva, R. C. P., Campelo, F. & Guimaraes, F. G. (2012). A multi-agent approach to the adaptation of migration topology in island model evolutionary algorithms, *Brazilian Symposium on Neural Networks, 2012, Curitiba* pp. 160–165.
- Lopes, R., Freitas, A., Silva, R. & Guimarães, F. (2012). Differential evolution and perceptron decision trees for classification tasks, in H. Yin, J. Costa & G. Barreto (eds), *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 550–557.
- Loukil, L., Mehdi, M., Melab, N., Talbi, E.-G. & Bouvry, P. (2009). A parallel hybrid genetic algorithm-simulated annealing for solving q3ap on computational grid, *Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing*, IEEE Computer Society, Washington, DC, USA, pp. 1–8.
- Luong, T. V., Melab, N. & Talbi, E.-G. (2010). Gpu-based island model for evolutionary algorithms, *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, pp. 1089–1096.

- Luque, G. & Alba, E. (2011). *Parallel Genetic Algorithms: Theory and Real World Applications*, Studies in Computational Intelligence, Springer.
- Maturana, J., Fialho, Á., Saubion, F., Schoenauer, M. & Sebag, M. (2009). Extreme Compass and Dynamic Multi-Armed Bandits for Adaptive Operator Selection, *IEEE Congress on Evolutionary Computation (CEC)*, Trondheim, Norvège, pp. 365–372.
- Mezura-Montes, E. n., Velázquez-Reyes, J. & Coello Coello, C. A. (2006). A comparative study of differential evolution variants for global optimization, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, ACM, New York, NY, USA, pp. 485–492.
- Miyagi, H., Tengan, T., Mohamed, S. & Nakamura, M. (2010). Migration effects on tree topology of parallel evolutionary computation, *TENCON, IEEE Region 10 International Conference*, pp. 1601–1606.
- Montgomery, D. C. (2006). *Design and Analysis of Experiments*, John Wiley & Sons.
- Olafsson, S. (2006). Metaheuristics, in S. G. Henderson & B. L. Nelson (eds), *Simulation*, Handbooks in Operations Research and Management Science, Elsevier, Amsterdam, The Netherlands, pp. 633–654. Chapter 21.
- Onwubolu, G. C. & Davendra, D. (2009). *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, 1st edn, Springer Publishing Company, Incorporated.
- Onwubolu, G. & Davendra, D. (2006). Scheduling flow shops using differential evolution algorithm, *European Journal of Operational Research* **171**: 674–692.
- Paterlini, S. & Krink, T. (2004). High performance clustering with differential evolution, *Congress on Evolutionary Computation CEC*, **2**: 2004–2011.
- Peng, L., Dai, G., Chen, F. & Liu, F. (2007). Study on application of multi-objective differential evolution algorithm in space rendezvous, *Proceedings of the 2nd international conference on Advances in computation and intelligence*, ISICA'07, Springer-Verlag, Berlin, Heidelberg, pp. 46–52.
- Pettey, C. C., Leuze, M. R. & Grefenstette, J. J. (1987). A parallel genetic algorithm, *International Computer Games Association*, pp. 155–161.

- Prado, R. S., Pedrosa Silva, R. C., Guimarães, F. G. & Neto, O. M. (2010a). A new differential evolution based metaheuristic for discrete optimization, *International Journal of Natural Computing Research* pp. 15–32.
- Prado, R. S., Pedrosa Silva, R. C., Guimarães, F. G. & Neto, O. M. (2010b). Using differential evolution for combinatorial optimization: A general approach, *Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 11–18.
- Price, K., Storn, R. M. & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Rao, S. (2009). *Engineering Optimization: Theory and Practice*, 4 edn, John Wiley & Sons.
- Rechenberg, I. (1973). *Evolutions strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog.
- Ruciński, M., Izzo, D. & Biscani, F. (2010). On the impact of the migration topology on the island model, *Parallel Computing* **36**: 555–571.
- Rudolph, G. (1991). Global optimization by means of distributed evolution strategies, *Proceedings of the 1st PPSN Conference*, Springer, pp. 209–213.
- Sai, H. A., Vinaya, B. A., Govardhan, A. & Satapathy, S. C. (2010). Data clustering using almost parameter free differential evolution technique, *International Journal of Computer Applications* **8**(13): 1–7.
- Salman, A., Engelbrecht, A. P. & Omran, M. G. (n.d.). Empirical analysis of self-adaptive differential evolution, *European Journal of Operational Research* **183**: 785–804.
- Schoenauer, M. & Michalewicz, Z. (1997). Evolutionary computation, *Control and Cybernetics* pp. 307–338.
- Silva, R. C. P., Lopes, R. A., Freitas, A. R. & Guimaraes, F. G. (2012). Performance comparison of parameter variation operators in self-adaptive differential evolution algorithms, *Brazilian Symposium on Neural Networks, Curitiba* pp. 148–153.

- Storn, R. & Price, K. (1995). Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical report*, International Computer Science Institute, Berkeley.
- Storn, R. & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Kluwer Academic Publishers, Hingham, MA, USA, pp. 341–359.
- Tan, P.-N., Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Tanese, R. (1989). Distributed genetic algorithms, *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 434–439.
- Tang, J., Lim, M.-H., Ong, Y.-S. & Er, M. J. (2004). Study of migration topology in island model parallel hybrid-ga for large scale quadratic assignment problems., *ICARCV*, IEEE, pp. 2286–2291.
- Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M. & Yang, Z. (2007). Benchmark functions for the CEC 2008 special session and competition on large scale global optimization, *Technical report*, Nature Inspired Computation and Applications Laboratory, USTC, China.
- Tasoulis, D. K., Pavlidis, N., Plagianakos, V. P. & Vrahatis, M. N. (2004). Parallel differential evolution, *In IEEE Congress on Evolutionary Computation CEC*, IEEE Press., pp. 2023–2029.
- Teo, J. (2006). Exploring dynamic self-adaptive populations in differential evolution, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, Vol. 10, Springer-Verlag, Berlin, Heidelberg, pp. 673–686.
- Tomassini, M. (1999). Parallel and Distributed Evolutionary Algorithms: A review, *Technical report*, Institute of Computer Science, University of Lausanne, Switzerland.
- Veronese, L. P. & Krohling, R. A. (2010). Differential evolution algorithm on the gpu with c-cuda, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*, IEEE, pp. 1–7.
- Viana, M. P. (2007). A metodologia das redes complexas para caracterização do sistema de havers, *Dissertação de Mestrado, Instituto de Física de São Carlos, Universidade de São Paulo*.

- Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning, *Machine Learning* **8**(3-4): 279–292.
- Weber, M., Tirronen, V. & Neri, F. (2010). Scale factor inheritance mechanism in distributed differential evolution, *Soft Computing* **14**: 1187–1207.
- Whitley, D., Rana, S. & Heckendorn, R. B. (1998). The island model genetic algorithm: On separability, population size and convergence, *Journal of Computing and Information Technology* pp. 33–47.
- Zamuda, A., Brest, J., Bošković, B. & Žumer, V. (2007). Differential Evolution for Multiobjective Optimization with Self Adaptation, *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*, IEEE Press, pp. 3617–3624.