

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Márcio Valério Weck Pereira

**OTIMIZAÇÃO MULTIMODAL PARA DOMÍNIO CONTÍNUO COM
HEURÍSTICAS DE AGRUPAMENTO ADAPTATIVO**

Florianópolis

2015

Márcio Valério Weck Pereira

**OTIMIZAÇÃO MULTIMODAL PARA DOMÍNIO CONTÍNUO COM
HEURÍSTICAS DE AGRUPAMENTO ADAPTATIVO**

Dissertação submetido ao Programa de Pós-Graduação em Ciências da Computação para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Mauro Roisenberg

Florianópolis

2015

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pereira, Márcio Valério Weck
Otimização Multimodal para Domínio Contínuo com Heurísticas
de Agrupamento Adaptativo / Márcio Valério Weck Pereira ;
orientador, Mauro Roisenberg - Florianópolis, SC, 2015.
92 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Otimização Multimodal. 3.
Niching. 4. CMA-ES. 5. Modelo Gaussiano. I. Roisenberg,
Mauro . II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Ciência da Computação. III.
Titulo.

Márcio Valério Weck Pereira

**OTIMIZAÇÃO MULTIMODAL PARA DOMÍNIO CONTÍNUO COM
HEURÍSTICAS DE AGRUPAMENTO ADAPTATIVO**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovado em sua forma final pelo Programa de Pós-Graduação em Ciências da Computação.

Florianópolis, 25 de fevereiro 2015.

Prof. Dr. Ronaldo dos Santos Mello
Coordenador
Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Dr. Mauro Roisenberg
Orientador

À minha família.

AGRADECIMENTOS

Agradeço primeiramente a meus pais, Márcio e Cristine, que estiveram sempre ao meu lado, dando todo o apoio que precisei. São eles que me incentivam a seguir meus sonhos e que me ensinaram a importância do esforço e do estudo para alcançá-los. Agradeço também à Mariluci, minha namorada, seu amor, compreensão e confiança foram essenciais nessa jornada. Agradeço ao meu tio e tia, Roberto e Leninha, que me receberam como um filho aqui em Florianópolis e sempre me apoiaram.

Agradeço ao prof. Mauro Roisenberg pela excelente orientação, pela amizade e pela confiança no meu trabalho. Pela motivação e pelos projetos nos quais trabalhamos juntos. Pela oportunidade de conhecer e visitar o Rio de Janeiro várias vezes!

Agradeço também ao Fernando, Cássio e Leandro, e aos demais membros do L3C, pelas discussões, ideias e parcerias. Aprendi muito trabalhando com essa equipe.

Agradeço também ao PPGCC (Programa de Pós-Graduação em Ciências da Computação). Principalmente à Katiana, que me ajudou várias vezes quando eu aparecia com problemas relacionados a documentos, inscrições e dissertação.

Agradeço também aos membros da banca pela participação e pelas observações feitas. Finalmente, agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela bolsa de mestrado.

“Essentially, all models are wrong, but some are useful.”

George Box

RESUMO

O crescente interesse nos métodos de otimização multimodal se deve a uma característica, quase que geral, dos problemas reais — a multimodalidade. Essa característica implica que o problema possui mais de uma solução ótima. Encontrar um conjunto de soluções *ótimas* é o objetivo dos métodos de otimização multimodal. O método apresentado neste trabalho, *Estratégia de Evolução Multimodal baseada em Multi-população*, ou NMESIS como será chamado devido a sua tradução para a língua inglesa Niche Multi-population Evolution Strategy with Improved Search, é um algoritmo de *niching* paralelo e explícito que utiliza como base a Adaptação da Matriz de Covariância. O método representa cada população como uma distribuição normal, o que permite utilizar técnicas destinadas à modelos de misturas gaussianas. Essa escolha ajuda a simplificar a parametrização, enquanto facilita o desenvolvimento de operadores robustos para troca de informação entre os nichos. O NMESIS foi avaliado através de um *benchmark*, utilizado em competições de algoritmos de *niching*, que contém 20 problemas de teste, especialmente concebidos para avaliação de métodos de otimização multimodal, e seu desempenho foi comparado a outros métodos no estado da arte como NMMSO, dADE e NEA2 (último vencedor do CEC 2013). Os resultados apresentados mostram que o NMESIS conseguiu encontrar mais soluções que os concorrentes. Outro fator positivo foi a consistência dos resultados, mesmo com o aumento da precisão.

Palavras-chave: otimização multimodal, *niching*, cma-es, agrupamento, modelo

ABSTRACT

The growing interest in multimodal optimization methods is motivated by an characteristic commonly found in real problems — multimodality. Find a set of optimal solutions is the target of multimodal optimization research. The method presented in this work, called Niche Multi-population Evolution Strategy with Improved Search (NMESIS), is a parallel *niching* method which is also explicit. Each niche is maintained by a CMA-ES instance. NMESIS abstracts the niche population as a Gaussian Mixture Model, allowing to use methods that are developed for classification and clustering. This helps to create robust operators to detect overlaps. Also, the abstraction allows a better communication mechanism between niches (migration). We apply a benchmark of 20 test functions, specially designed for multimodal optimization evaluation, and compare the performance with state-of-the-art methods. Finally we discuss the results and show that the proposed approach can reach better and stable results even in high-dimensional spaces.

Keywords: multimodal optimization, niching, cma-es, clustering model

LISTA DE FIGURAS

<p>Figura 1 À esquerda temos uma função unimodal, com um valor de ótimo bem destacado; Já à direita, temos uma função simétrica e multimodal (6 soluções ótimas).</p> <p>Figura 2 Desempenho dos métodos que foram avaliados pelo CEC 2013. Em azul é a porcentagem de métodos que acertaram mais de 80% das soluções. Em verde estão aqueles que acertaram mais de 50% e em roxo aqueles que acertaram mais 20%</p> <p>Figura 3 O modelo geométrico de Fisher é um modelo evolucionário dos efeitos causados por mutações espontâneas na aptidão do indivíduo.</p> <p>Figura 4 Exemplo de como a Equação 2.3 afeta a matriz de covariância. Esquerda: $z^{(g+1)}$. Meio: $D^{(g)}z^{(g+1)}$. Direita: $B^{(g)}D^{(g)}z^{(g+1)}$. Figura retirada de (HANSEN, 2005)</p> <p>Figura 5 Dois nichos representados por suas respectivas distribuições (\mathcal{N}^D).</p> <p>Figura 6 Modelo biogeográfico de ilhas (MACARTHUR, 1967).</p> <p>Figura 7 Histogramas de populações com 5, 10 e 20 dimensões. A linha que cruza o eixo y mostra o θ escolhido utilizando $\tau = 0.95$ na inversa da função de probabilidade acumulada da χ^2.</p> <p>Figura 8 À esquerda, o agrupamento feito através do NBC somente; À direita, o agrupamento utilizando o NBC+DM com 1 ponto interno.</p> <p>Figura 9 Arquitetura do NMESIS. Um controle central que aplica os operadores enquanto cada nicho executa suas operações de evolução utilizando o CMA-ES.</p> <p>Figura 10 Quatro gerações capturadas durante a execução do método. Na ordem a 4, 16, 24 e 40. Os pontos pretos indicam as soluções enquanto os pontos vermelhos indicam o centroide do nicho. Na última, as estrelas vermelhas indicam o local dos ótimos globais. Todos foram encontrados.</p> <p>Figura 11 Razão de Acerto (RA) para o <i>benchmark</i> CEC 2013. (a) RA de cada algoritmo para precisão de $1e - 01$. (b) Idem para a precisão de $1e - 05$.</p> <p>Figura 12 Taxa de Sucesso (TS) para o <i>benchmark</i> CEC 2013. (a) TS de cada algoritmo para precisão de $1e - 01$. (b) Idem para a precisão de $1e - 05$.</p>	<p>26</p> <p>29</p> <p>35</p> <p>39</p> <p>51</p> <p>52</p> <p>55</p> <p>59</p> <p>60</p> <p>66</p> <p>74</p> <p>75</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------

LISTA DE TABELAS

Tabela 1	Comparação entre PSO e CMA-ES sob as questões de invariância do método e mal-condicionamento do problema.	40
Tabela 2	Parâmetros das funções usados para medir a performance. . . .	73
Tabela 3	Configuração do NMESIS	74
Tabela 4	Razão de Acertos (RA) e Taxa de Sucesso (TS) do NMESIS. . . .	76
Tabela 5	Taxa de Convergência para os métodos NMESIS, NMMSO, dADE/nrand/1, TN-CMA-ES e BIPOP-CMA-ES. Com precisão especificada de $1e-01$. Os melhores resultados estão destacados.	77
Tabela 6	Taxa de Convergência para os métodos NMESIS, NMMSO, dADE/nrand/1, TN-CMA-ES e BIPOP-CMA-ES. Com precisão especificada de $1e-04$. Os melhores resultados estão destacados.	78

LISTA DE ABREVIATURAS E SIGLAS

NEA2	<i>Niching Evolutionary Algorithm 2</i>	27
NMMSO	<i>Niching Migratory Multi-Swarm Optimiser</i>	27
dADE	<i>dynamic Archive Differential Evolution</i>	28
GA	<i>Genetic Algorithm</i>	31
EA	<i>Evolutionary Algorithms</i>	31
ES	<i>Evolution Strategy</i>	32
NSGA-II	<i>Non- dominated Sorting Genetic Algorithms-II</i>	32
CMA	<i>Covariance Matrix Adaptation</i>	34
CMA-ES	Algoritmo combinado de ES com a CMA.	35
PSO	<i>Particle Swarm Optimization</i>	37
SN	<i>Sequential Niching</i>	42
PN	<i>Parallel Niching</i>	42
HV	<i>Hill-Valley</i>	45
NBC	<i>Nearest Better Clustering</i>	46
GCES	<i>Gaussian Classifier Evolution Strategy</i>	46
CEC	<i>IEEE Congress of Evolutionary Computing</i>	47
NMESIS	<i>Niched Multi-population Evolution Strategy with Improved Search</i>	50
GMM	<i>Gaussian Mixture Model</i>	51
NBC	<i>Nearest Better Clustering</i>	57
LHS	<i>Latin Hypercube Sampling</i>	63
RA	Razão de Acertos	71
NR	Número de rodadas (execuções).	71
NPF_i	Número de ótimos encontrados na i -ésima rodada.	71
NKP	Número de ótimos globais conhecidos.	71
NSR	número de execuções bem sucedidas.	71
TS	Taxa de Sucesso	71
VMC	Média do número de funções avaliados para cada execução do <i>benchmark</i>	71
NFA_i	Número de funções avaliadas na i -ésima execução do <i>benchmark</i>	71
EM	<i>Expectation Maximization</i>	80

LISTA DE SÍMBOLOS

D	Número de dimensões	26
t	Instante de tempo.	34
(g)	Geração ou iteração em que o método se encontra.	35
$\langle \mathbf{x} \rangle^{(g)}$	Centroide da geração g do CMA-ES	36
$\sigma^{(g)}$	Tamanho do passo global na geração g do CMA-ES.	36
\mathbf{B}	Matriz de autovetores do CMA-ES.	36
\mathbf{D}	Matriz diagonal de autovalores do CMA-ES.	36
$\mathbf{C}^{(g)}$	Matriz de covariância da geração g do CMA-ES	37
$\mathbf{B}^{(g)}$	Matriz de autovetores da geração g do CMA-ES.	37
$D^{(g)}$	Matriz de autovalores da geração g do CMA-ES.	37
$\kappa(\mathbf{A})$	Função que retorna o número de condicionamento da matriz \mathbf{A}	39
Λ	Tamanho da população total	50
Φ	Número de buscas (instâncias do CMA-ES) simultâneas	50
d_{KL}	<i>Divergência de Kullback-Leibler</i>	52
Σ_k	Matriz de covariância da k -ésima distribuição.	52
md_{AB}	Distância de Mahalanobis entre a distribuição A e B.	53
$g_k(\vec{x})$	Modelo gaussiano da RDB.	53
\vec{x}	Vetor da solução amostrada.	53
\vec{m}_k	Centroide da distribuição.	53
τ	Quantil definido pela Regra de Decisão Bayesiana.	54
θ	Limiar da distância de Mahalanobis	54
χ^2	Distribuição Chi-Quadrado	54
\mathbf{F}^{-1}	Função de probabilidade acumulada.	54
P	População de soluções.	58
μ_{nb}	Média das distâncias do grafo do NBC.	58
$n_{int_{ini}}$	Número de pontos interiores usados pelo Hill- Valley.	58
q_k	Tupla de valores que representa um nicho.	59
λ_k	Tamanho da população do k -ésimo nicho.	59
d_{nb}^k	Distância calculada usando a definição do “melhor-mais- próximo”.	60
$lhs(\lambda)$	Amostragem geométrica usando hipercubo latino.	63

SUMÁRIO

1	INTRODUÇÃO	25
1.1	JUSTIFICATIVA	28
1.2	OBJETIVO	29
1.2.1	Objetivos Específicos	29
1.3	ORGANIZAÇÃO DO TRABALHO	30
2	OTIMIZAÇÃO MULTIMODAL CONTÍNUA	31
2.1	OTIMIZAÇÃO GLOBAL CONTÍNUA	31
2.1.1	Algoritmos Genéticos	32
2.2	ESTRATÉGIAS EVOLUTIVA	33
2.2.1	Adaptação da Matriz de Covariância	34
2.2.1.1	Maximização do Sucesso e Caminhos Evolutivos	35
2.2.1.2	Algoritmo	36
2.2.2	Funções Mal-Condicionadas e não-separáveis	37
2.3	OTIMIZAÇÃO MULTIMODAL	40
2.4	<i>NICHING</i>	41
2.4.1	<i>Niching</i> Sequencial	42
2.5	<i>NICHING</i> PARALELO	42
2.5.1	Técnicas Clássicas	43
2.5.2	Técnicas de <i>Niching</i> explícito	44
2.6	TRABALHOS RELACIONADOS	46
2.7	DISCUSSÃO	47
3	PROPOSTA	49
3.1	MODELAGEM DOS NICHOS VIA MISTURAS GAUSSIANAS	51
3.1.1	Função de Similaridade	52
3.2	ALGORITMO	56
3.2.1	População Inicial	56
3.2.2	<i>Niching</i>	58
3.2.3	Validação do Nicho	65
3.3	DISCUSSÃO	67
4	EXPERIMENTOS	69
4.1	<i>BENCHMARK</i> CEC 2013	69
4.1.1	Métricas de Performance	70
4.1.2	Configuração do Experimento	71
4.1.3	Resultados	72
4.2	DISCUSSÃO	77
5	CONCLUSÃO	79
	REFERÊNCIAS	83

1 INTRODUÇÃO

O tema *otimização* envolve uma mistura de heurísticas e rigor, teórico e experimental, que permite aplicações em diversas áreas do conhecimento. Segundo Fletcher (2013), a otimização pode ser definida como a ciência de se obter as “melhores” soluções para problemas matemáticos. Tais problemas normalmente modelam situações reais e são designados como *função objetivo* do problema. Esta ciência envolve estudar a estrutura dos problemas, os critérios que definem uma solução “ótima”, a determinação do algoritmo e experimentos computacionais.

Basicamente, qualquer atividade que possua um certo grau de complexidade permite algum tipo de otimização. A aplicabilidade dos métodos de otimização é tão vasta, que praticamente envolve todas as área em que algum tipo de informação é processada (engenharia, matemática, economia, etc.) (FLETCHER, 2013). Algumas das principais áreas de aplicação são: *design* de turbinas e estruturas de aviões; *design* de estruturas usadas na construção civil; logística — alocação de recursos, agendamento e combinação (NOREIKIS; BUTKUS; NURMINEN, 2014); inversão de formas de onda sísmica (KOPER; WYSESSION; WIENS, 1999); gerenciamento de combustível (TURINSKY, 2010) e otimização do *design* de reatores nucleares (PEREIRA; SCHIRRU; MARTINEZ, 1999).

No entanto, os métodos de otimização geralmente tem como objetivo encontrar a melhor solução dado uma função objetivo multidimensional. Para algumas aplicações esse comportamento pode não ser ideal, uma vez que apenas uma solução é encontrada (DAS et al., 2011). A possibilidade de alternar entre soluções sem perder a qualidade normalmente é um requisito desejado em problemas de *design* e planejamento (PEREIRA; SCHIRRU; MARTINEZ, 1999; NOREIKIS; BUTKUS; NURMINEN, 2014). Outra aplicação é utilizar diferentes soluções para calcular estimativas (e.g. incerteza) (SAMBRIDGE, 2013).

A Figura 1 ilustra a diferença entre um problema unimodal (a melhor solução é única), um multimodal real (5 soluções). No primeiro caso, a função realmente tem apenas uma solução ótima global, i.e. ao aplicar essa solução na função objetivo obtêm-se o resultado máximo da função. Já no segundo caso, as cinco soluções indicadas retornam o valor máximo. Problemas como o segundo caso, são considerados problemas multimodais. Entretanto, problemas aonde encontrar as soluções que são máximas dentro de sua vizinhança, chamadas de ótimos locais (diferentemente dos ótimos globais), também podem ser considerados multimodais, uma vez que mais de uma solução deve ser encontrada. Em suma, problemas multimodais envolvem obter mais de um

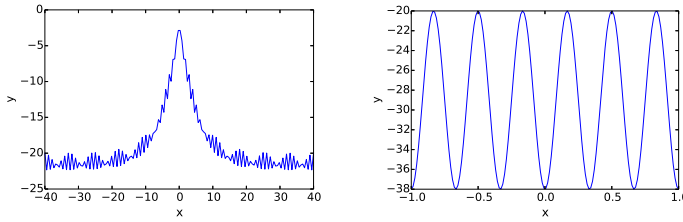


Figura 1 – À esquerda temos uma função unimodal, com um valor de ótimo bem destacado; Já à direita, temos uma função simétrica e multimodal (6 soluções ótimas).

ótimo local (GOLDBERG; RICHARDSON, 1987; MAHFOUD, 1995a; DAS et al., 2011).

Embora várias soluções possam ser obtidas executando repetidamente métodos de otimização unimodal com diferentes soluções iniciais (DAS et al., 2011; PREUSS; STOEAN; STOEAN, 2011), não há nenhuma garantia de que soluções distintas serão encontradas. Em Vasco, Jr e Majer (1996) essa estratégia foi utilizada para obter diversos modelos de um problema inverso de tomografia sísmica. Métodos mais recentes oferecem uma alternativa eficiente para fornecer soluções distintas para um problema de otimização. Tais métodos utilizam o conceito de *niching* (DEB; GOLDBERG, 1989; DAS et al., 2011) para construir heurísticas capazes de encontrar e manter um conjunto de soluções (BEASLEY; BULL; MARTIN, 1993; MAHFOUD, 1995a).

Os métodos de *niching* são mais complexos e nem sempre adequados para alguns problemas multimodais. Por exemplo, em problemas maiores, aqueles baseados em medidas de distância podem falhar uma vez que, para $D > 5$ (aonde D é o número de dimensões), o contraste entre as distâncias reduz dramaticamente, e quando $D \rightarrow \infty$ os pontos no espaço de busca estarão, efetivamente, separados pela mesma distância (PREUSS; STOEAN; STOEAN, 2011). Entretanto existem muitos problemas com dimensionalidade moderada, de 5 a 20 dimensões, que podem ser resolvidos por métodos de *niching* (PREUSS; STOEAN; STOEAN, 2011; DAS et al., 2011). Segundo Preuss, Stoean e Stoean (2011), os métodos de *niching* eram avaliados utilizando problemas que não refletiam a aplicação prática desses métodos, com *benchmarks* envolvendo funções com no máximo três dimensões.

A necessidade de um *benchmark* unificado e com problemas mais desafiadores culminou na criação de um conjunto de funções objetivo com um espectro maior de problemas (LI; ENGELBRECHT; EPITROPAKIS, 2013a).

Neste trabalho, esse conjunto será designado por *benchmark* CEC 2013 ou apenas *CEC 2013*, em referência ao Congresso de Computação Evolucionária (*IEEE Congress on Evolutionary Computation — CEC*) que utilizou esse conjunto de funções como *benchmark* para sua competição de algoritmos de otimização multimodal.

O presente trabalho tem como base os métodos de *niching* paralelos (DAS et al., 2011) para problemas contínuos de maximização:

$$\arg \max_{\vec{x} \in \mathbb{R}^D} F(\vec{x}) \in \mathbb{R},$$

onde $F(\vec{x})$ é a função objetivo. Entretanto, a utilização das técnicas descritas neste trabalho para problemas de minimização requer apenas mudanças triviais, como qualquer outro método de otimização. Portanto, para fins de simplificação, todos os problemas apresentados ao longo do texto serão problemas de maximização. Tanto os problemas como o protocolo de avaliação são os mesmos empregados no CEC 2013.

O *benchmark* mais atual é o CEC 2013, que inclusive foi utilizado na competição do CEC 2015. Sendo assim, qualquer algoritmo avaliado usando o mesmo protocolo estabelecido pelo CEC 2013 poderá ser facilmente comparado com o algoritmo proposto neste trabalho.

Os métodos estado-da-arte com melhor desempenho e, principalmente, que obtiveram bons resultados para $D > 5$ são: NEA2 (PREUSS, 2012), que utiliza uma técnica sequencial de *niching* juntamente com um algoritmo de inicialização especializado para detectar regiões de atração; o NMMSO (FIELDSEND, 2014), utiliza uma estratégia de subpopulações para manter a diversidade da população e métricas baseadas em topologia; e TN-CMA-ES (PEREIRA; NETO; ROISENBERG, 2014), que também utiliza subpopulações e informação topológica, porém com uma inicialização semelhante ao NEA2. Na competição realizada em 2013 (LI; ENGELBRECHT; EPITROPAKIS, 2013a) o vencedor foi o NEA2. Já em Fieldsend (2014) o autor argumenta que o seu método havia superado os demais (incluindo o NEA2) na performance geral, no entanto ainda tinha um desempenho inferior ou similar nas funções para $D > 5$. Dessa maneira, é importante perceber que aplicações como Vasco, Jr e Majer (1996), Sacco, Henderson e Rios-Coelho (2014) e Noreikis, Butkus e Nurminen (2014), estão limitadas aos problemas de escalabilidade dos métodos de *niching*.

1.1 JUSTIFICATIVA

Sendo uma tarefa computacionalmente cara, a otimização multimodal requer alto desempenho para que se mantenha viável em aplicações reais (DAS et al., 2011). Particularmente, aplicações dinâmicas em tempo-real tornam esse requisito ainda mais severo. É importante que o método seja escalável com relação ao tamanho do problema (número de dimensões).

Segundo Fieldsend (2014), um método de otimização multimodal deve possuir determinadas características para ser efetivo. Primeiramente, o método precisa ser dinâmico com relação ao número de soluções mantidas. Isso significa que o método não pode depender de suposições iniciais quanto ao número de soluções que devem se encontradas. A segunda característica, é adaptar-se automaticamente. Tal propriedade permite que o algoritmo tenha um desempenho uniforme em diferentes problemas. A adaptação envolve tornar dinâmicos os parâmetros do método, diminuindo a dependência do estado inicial do algoritmo.

Outra propriedade importante é incorporar uma otimização local à estrutura do método. Essa estratégia é bastante utilizada como forma de tentar resolver o conflito entre “exploração e exploração”¹ (ALBA; DORRONSORO, 2005). Métodos que utilizam a combinação de algoritmos de otimização global e local são conhecidos como algoritmos meméticos (*Memetic algorithms*) (MOSCATO et al., 1989). Embora o conceito esteja relacionado, o estudo dos algoritmos meméticos vai além do escopo deste trabalho e da propriedade apresentada. Na otimização multimodal, o foco desta propriedade é corroborar para que o algoritmo possua uma “visão” local da região encontrada, facilitando a sua manutenção uma vez que influências externas (outras regiões de atração) são eliminadas.

Em Fieldsend (2014), o autor ao comparar o seu método, NMMSO, com outros algoritmos como o NEA2 (PREUSS, 2012) e o dADE (EPITROPAKIS; LI; BURKE, 2013), ressalta que esses algoritmos incorporam as propriedades descritas para um otimizador multimodal efetivo. No entanto, os indicadores do CEC 2013 mostram que a performance desses métodos, em problemas com $D > 5$, ainda está aquém do desejado para aplicá-los em situações reais, como mostra o gráfico da Figura 2. O gráfico apresenta uma visão condensada do desempenho dos métodos que foram avaliados pelo CEC 2013. Foi feita uma contagem dos métodos que obtiveram taxas de acerto maiores que 20%, 50%, 80%, para que fosse possível observar como os métodos, em conjunto, se comportaram em cada função. Claramente, é possível ver a necessidade de aprimorar o desempenho em problemas mais complexos e

¹ Expressão mais conhecida em inglês: *exploration vs exploitation*

principalmente, com mais dimensões.

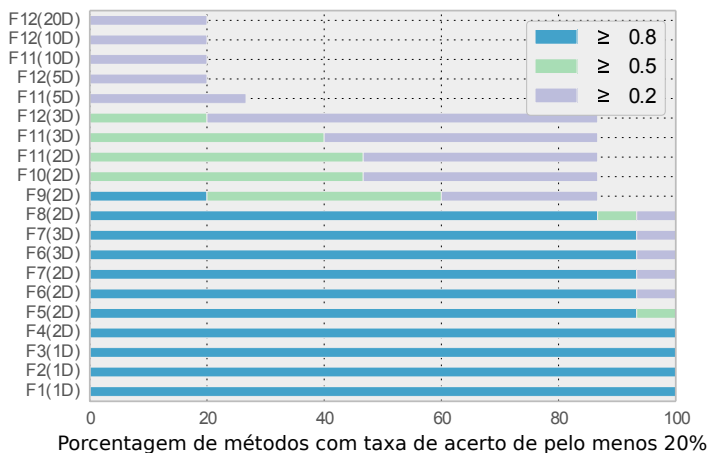


Figura 2 – Desempenho dos métodos que foram avaliados pelo CEC 2013. Em azul é a porcentagem de métodos que acertaram mais de 80% das soluções. Em verde estão aqueles que acertaram mais de 50% e em roxo aqueles que acertaram mais 20%

1.2 OBJETIVO

O objetivo deste trabalho é obter um método de *niching* mais eficiente em problemas complexos e de tamanho moderado, utilizando como comparação um *benchmark* específico para otimização multimodal.

1.2.1 Objetivos Específicos

1. Revisar as técnicas de *niching* disponíveis e mapear heurísticas eficazes.
2. Propor e avaliar operadores alternativos (*i.e.*, não encontrados na literatura) para *niching* visando a manutenção da diversidade dos indivíduos da população.
3. Projetar e implementar um algoritmo seguindo o conceito de *niching*

para otimização multimodal.

4. Realizar experimentos para demonstrar a capacidade do método e dos operadores propostos.

1.3 ORGANIZAÇÃO DO TRABALHO

Esse trabalho está organizado da seguinte forma: no próximo capítulo é feita uma revisão dos conceitos básicos de otimização global, multimodalidade e estratégias de evolução. No mesmo capítulo são introduzidas as técnicas de otimização multimodal, principalmente os métodos de *niching*. Por fim, são apresentados alguns trabalhos relacionados ao tema da dissertação. No Capítulo 3, o método proposto, o NMESIS, é apresentado. Na sequência, o Capítulo 4 trás os resultados do método e a comparação com outros métodos no estado da arte. E finalmente, o Capítulo 5 apresenta as considerações finais sobre o trabalho.

2 OTIMIZAÇÃO MULTIMODAL CONTÍNUA

Otimização é a tarefa de selecionar a melhor alternativa dentre um conjunto de opções, onde cada decisão deve ser tomada para otimizar um ou mais objetivos, de acordo com um conjunto de restrições (BAGHEL; AGRAWAL; SILAKARI, 2012). Problemas de otimização podem surgir nas mais diversas disciplinas, tais como engenharia, *design*, sistemas de produção, economia, biologia e etc. Tal abrangência prática dos problemas de otimização implica na necessidade de métodos robustos e eficientes computacionalmente para que modelos matemáticos de médio e grande porte possam ser resolvidos (BAGHEL; AGRAWAL; SILAKARI, 2012).

Este Capítulo tem a intenção de fazer uma revisão das metaheurísticas e suas aplicações em problemas de otimização. Geralmente os algoritmos de otimização multimodal são desenvolvidos ao redor de outros métodos já estabelecidos, através da colaboração com novas heurísticas ou modificações nos operadores originais, permitindo a manutenção da diversidade de soluções.

2.1 OTIMIZAÇÃO GLOBAL CONTÍNUA

A *otimização global contínua* designa uma classe de métodos desenvolvida para explorar o espaço de busca exaustivamente, porém ao contrário de uma busca aleatória, heurísticas são utilizadas em diferentes etapas para acelerar a convergência do método para um ótimo global. Os métodos dessa classe devem codificar suas soluções no domínio contínuo, \mathbb{R}^D aonde D é o número de dimensões (GOLDBERG, 1989; HOLLAND, 1975). Geralmente os métodos de otimização global são desenvolvidos como “caixa-preta”, eliminando a necessidade de calcular a derivada da função objetivo, exatamente como se a função objetivo fosse uma “caixa” inacessível para o método e do qual este não pode tirar nenhuma informação. Essa restrição motiva os pesquisadores a buscar soluções em processos físicos e biológicos, conhecidas como *metaheurísticas*, alguns dos exemplos mais famosos são: Algoritmo Genético (*Genetic Algorithm - GA*) (HOLLAND, 1975), Têmpera Simulada (KIRKPATRICK, 1984), Busca Tabu (GLOVER, 1989), Enxame de Partículas (KENNEDY, 2010), entre outros. Nenhuma dessas técnicas garante que a melhor solução (ótimo global) será encontrada, no entanto, existe uma grande probabilidade de que a solução obtida seja uma boa solução e que o tempo para obtê-la é menor que uma busca aleatória.

Os *Algoritmos Evolucionários (Evolutionary Algorithms - EA)* (GOLDBERG, 1989; MÜHLENBEIN; MAHNIG, 2001) são métodos que comparti-

lham operadores relacionados a evolução de organismos e comunidades, sendo métodos que inerentemente utilizam uma população de soluções para capturar informações do espaço de busca e “evoluir” seus indivíduos (MÜHLENBEIN; MAHNIG, 2001).

Atualmente, as Estratégias Evolutivas (*Evolution Strategy* - ES) vem ganhando mais atenção e seus algoritmos tem demonstrado bons resultados ao longo dos anos (WHITLEY et al., 1996; HANSEN, 2009; HANSEN et al., 2010; LIANG et al., 2014). Juntamente com operadores de mutação capazes de se auto adaptar (BEYER; SCHWEFEL, 2002; HANSEN; OSTERMEIER, 2001) tornam a ES a opção mais atrativa dentre os métodos atuais. Portanto, a Subseção a seguir apenas revisita o GA, enquanto a Seção 2.2 irá apresentar uma visão mais aprofundada das Estratégias Evolutivas e a variante utilizada neste trabalho.

2.1.1 Algoritmos Genéticos

Os Algoritmos Genéticos (GAs) (HOLLAND, 1975; GOLDBERG, 1989) são o cerne da Computação Evolucionária, tanto por serem definidos sobre uma das abstrações que mais está associada ao processo evolutivo (os genes ou genética), como pela simplicidade que os mecanismos evolutivos se combinam no algoritmo. Esses fatores tornam o GA uma base para desenvolver outras técnicas de otimização global (MAHFOUD, 1995a). Basicamente, um dos pontos chave do GA é a forma como as soluções são representadas, possibilitando ao algoritmo explorar corretamente o espaço de busca e convergir em áreas de atração realmente promissoras (evitando ótimos locais). É o caso do Problema do Caixeiro Viajante, em que uma solução é codificada como sendo a sequência de cidades a serem visitadas. Ainda é possível adaptar os operadores de acordo com o problema, como por exemplo, no TSP não é permitido repetir cidades, portanto os operadores de mutação e cruzamento devem ser desenvolvidos especificamente para esse problema.

Embora a pesquisa em torno dos GAs terem diminuído, principalmente em problemas de otimização unimodais onde as Estratégias Evolutivas tem ganhado maior destaque, algumas variantes tem se destacado em problemas multi-objetivo (*Non-dominated Sorting Genetic Algorithms-II* - NSGA-II) (DEB et al., 2002). Demonstrando que o GA é um modelo genérico e extremamente flexível para a construção de novas abordagens de otimização.

2.2 ESTRATÉGIAS EVOLUTIVA

Introduzida por Ingo Rechenberg e Hans-Paul Schwefel (RECHENBERG, 1989; BEYER; SCHWEFEL, 2002). A *estratégia evolutiva*, constrói sua população amostrando novas soluções candidatas de acordo com uma distribuição normal multivariada em \mathbb{R}^D (\mathcal{N}^D). As mutações são aplicadas somando uma perturbação com média zero (um vetor aleatório) em cada solução amostrada (ORR, 2000). As dependências entre as variáveis da distribuição são representadas por uma matriz de covariância. Por definição, a covariância é a medida numérica de quanto duas variáveis mudam juntas. Para problemas com mais de uma dimensão, a covariância é representada por uma matriz simétrica e positiva definida. O procedimento descrito é dado pela equação:

$$\begin{aligned} y_i^{(g+1)} &= \mathcal{N}(x^{(g)}, (\sigma^{(g)})^2 C^{(g)}) \\ &= x^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}), \quad i = 0, 1, 2, \dots, N, \end{aligned} \quad (2.1)$$

onde g é geração, $x^{(g)}$ é o indivíduo selecionado como centroide, $\sigma^{(g)}$ é a quantidade de mutação a ser aplicada (passo evolutivo ou passo de mutação) e $C^{(g)}$ é a matriz de covariância. A nova solução $y_i^{(g+1)}$ fará parte da próxima população. A solução $x^{(g)}$ pode ser tanto escolhida da população como ser a média dos melhores indivíduos. Além disso, a forma como a solução escolhida é utilizada varia de acordo com o objetivo do método. Para identificar essas estratégias existe uma notação especial $(\mu; \lambda) - ES$:

1. $(\mu, \lambda) - ES$ utiliza μ indivíduos para definir o centro da distribuição e gera λ novos indivíduos;
2. $(\mu + \lambda) - ES$ é um modelo elitista que preserva os melhores;
3. $(1, \lambda) - ES$ seleciona o melhor e usa como centro para criar λ novos indivíduos;
4. $(1 + \lambda) - ES$ elitista mantendo alguma solução da população anterior (o melhor ou a média) na nova população;

As duas primeiras são mais robustas, uma vez que possuem mais informação para decidir qual será o próximo centro, entretanto são métodos computacionalmente mais caros. Os dois últimos são mais eficientes porém podem ter dificuldades em funções não-convexas e na presença de ruído ou armadilhas (ótimos locais indesejados) (HANSEN; MÜLLER; KOUMOUTSAKOS, 2003; LOSHCHILOV, 2014).

Atualmente, o foco das pesquisas sobre Estratégias Evolutivas é a adaptação dos parâmetros, como a matriz C , o tamanho do passo σ e o tamanho da população (LOSHCHILOV et al., 2014). O σ precisa ser sempre corrigido a tempo, ou seja, quando mais rápido o algoritmo aprender sobre o problema, melhor será o tamanho do passo escolhido (HANSEN; MÜLLER; KOUMOUTSAKOS, 2003). Porém, a adaptatividade é uma propriedade que, geralmente, é mais custosa pois envolve armazenar informação e aprender sobre essa informação armazenada. Alguns estudos como Loshchilov (2014) e Loshchilov et al. (2014) buscam reduzir a complexidade do aprendizado através da compactação da informação armazenada pela matriz de covariância.

2.2.1 Adaptação da Matriz de Covariância

Os métodos de otimização iterativos tem como base dois conceitos importantes: a “direção” em que uma modificação (mutação) é aplicada e a quantidade de informação que representa essa modificação. Esses dois conceitos são chamados de *gradiente* da função e *tamanho do passo* do algoritmo. A necessidade de compreender e obter valores ótimos para essas propriedades já era objeto de estudo na publicação de “The Genetical Theory of Natural Selection” por Ronald Fisher (FISHER, 1930). O Modelo Geométrico de Fisher é um modelo evolucionário que descreve o efeito causado pelas mutações na aptidão de um organismo. O modelo é ilustrado pela Figura 3, por ser um modelo 2D, significa que o organismo tem duas características genéticas (traços) em que **o centro é o estado de aptidão máxima**. Com relação ao *gradiente*, mutações que deslocam o indivíduo para dentro do círculo são benéficas, enquanto movimentos para fora não. Já o *tamanho do passo* influencia na velocidade com a qual o indivíduo alcança a solução ótima, entretanto essa não é uma relação linear. Por exemplo, na Figura 3 uma das setas atravessa completamente o círculo interno, indicando que o passo foi demasiado longo. Esse indivíduo provavelmente terá dificuldades para convergir, já que independente de quão próximo o passo t chegue próximo do centro o passo $t + 1$ o levará para longe novamente. E se por acaso o passo for pequeno, pode ser que ele seja tão pequeno que torne a convergência extremamente demorada (FISHER, 1930). Portanto, em algoritmo de otimização genéricos iterativos, a solução é utilizar algum mecanismo de adaptação para atualizar os parâmetros a cada iteração.

O mecanismo desenvolvido por Hansen e Ostermeier (1996), a adaptação da Matriz de Covariância (*Covariance Matrix Adaptation* - CMA), ataca o problema da adaptação aproveitando a forma como as Estratégias Evolutivas aplicam a mutação. Como foi visto no início da seção, a mutação é definida

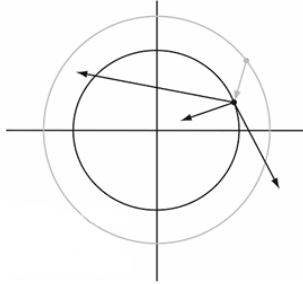


Figura 3 – O modelo geométrico de Fisher é um modelo evolucionário dos efeitos causados por mutações espontâneas na aptidão do indivíduo.

pela Equação 2.1, que para fins de simplificação foi replicada abaixo:

$$y_i^{(g+1)} = x_i^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}), \quad i = 0, 1, 2, \dots, N,$$

como o próprio nome sugere, o método adapta a matriz de covariância ($\frac{n(n-1)}{2}$ parâmetros para atualizar) na intenção de aprender qual é a derivada de segunda ordem da função objetivo subjacente, de forma similar à aproximação da matriz Hessiana usada por métodos *Quasi-Newtonianos* na otimização clássica (MORÉ, 1978). A diferença é que ao contrário desses métodos, não há nenhuma suposição sobre a função objetivo, apenas o *ranking* das soluções é explorado para aprender a distribuição das amostras. O CMA também adapta o tamanho do passo, de maneira que o algoritmo consiga ajustar o parâmetro rapidamente ao longo da execução do método. Quanto menos iterações o algoritmo leva para ajustar mais eficiente fica a convergência.

Essa técnica deu origem a uma Estratégia Evolutiva (ES) completa, o CMA-ES. Um método de otimização global para problemas de otimização “caixa-preta”, não-lineares, não-convexos e mal-condicionados (HANSEN; OSTERMEIER, 2001; HANSEN; MÜLLER; KOUMOUTSAKOS, 2003; HANSEN et al., 2010).

2.2.1.1 Maximização do Sucesso e Caminhos Evolutivos

O CMA-ES (HANSEN; OSTERMEIER, 2001; HANSEN; MÜLLER; KOUMOUTSAKOS, 2003) utiliza dois conceitos importantes: O primeiro é **princípio da máxima verossimilhança**, em que a média da distribuição é calculada de forma que a probabilidade dos passos anteriores que foram bem

sucedidos seja maximizada. Como não há informação suficiente no início, a matriz de covariância é atualizada incrementalmente para que, da mesma maneira que a média, a probabilidade dos passos bem sucedidos aumente. O custo desse aprendizado é ter que conduzir uma análise dos componentes principais iterativa dos movimentos bem sucedidos enquanto mantêm todos os eixos principais.

O segundo é o conceito de caminho evolutivo ou caminho de busca. Esses “caminhos” são vetores que armazenam informação sobre a correlação entre gerações consecutivas. Quando gerações consecutivas movem-se na mesma direção, a influência do caminho evolutivo aumenta, acelerando o aprendizado de direções favoráveis. Também pode ser utilizado para controlar o tamanho do passo, de forma independente da matriz de covariância, conhecido como *adaptação acumulativa do tamanho do passo* (HANSEN; MüLLER; KOUMOUTSAKOS, 2003; HANSEN, 2005). Esse controle é importante para prevenir convergência prematura do método ao mesmo tempo que permite convergir rapidamente para uma solução ótima (HANSEN, 2005). Basicamente, essa técnica tenta superar o problema do tamanho do passo, exposto anteriormente usando o modelo de Fisher.

2.2.1.2 Algoritmo

A variação mais comum e que segue todos os conceitos descritos aqui, é o (μ, λ) -CMA-ES. Essa é a configuração utilizada como base para construir o algoritmo proposto neste trabalho. O Algoritmo 1 esboça o funcionamento do CMA-ES. Os símbolos utilizados são:

$\mathbf{x}_i^{(g+1)} \in \mathbb{R}^D$, i -ésimo indivíduos da geração $g + 1$.

$\langle \mathbf{x} \rangle^{(g)}$ a média aritmética ponderada dos μ melhores indivíduos da geração g .

$\sigma^{(g)} \in \mathbb{R}^+$ tamanho do passo global.

$\mu_{eff} = \left(\sum_{i=0}^{\mu} w_i^2 \right)^{-1}$ é a variância efetiva da seleção, sendo $1 \leq \mu_{eff} \leq \mu$. w_i designa os pesos aplicados na média ponderada usada para calcular o centroide.

$\mathbf{B}^{(g)}$ matriz $N \times N$ ortogonal que determina as coordenadas do sistema. As colunas são os autovetores normalizados da matriz de covariância $\mathbf{C}^{(g)}$.

$\mathbf{D}^{(g)}$ matriz diagonal da qual os elementos da diagonal são as raízes quadradas dos autovalores de $\mathbf{C}^{(g)}$.

$\mathbf{z}_i^{(g+1)} \in \mathbb{R}^n$ amostra de uma distribuição $\mathcal{N}(0, I)$ (distribuição normal multivariada padrão).

$\mathbf{C}^{(g)}$ matriz de covariância $N \times N$, $\mathbf{B}^{(g)}$ e $D^{(g)}$ são determinados pela decomposição¹ de $\mathbf{C}^{(g)}$, que é $\mathbf{C}^{(g)} = \mathbf{B}^{(g)} D^{(g)} (\mathbf{B}^{(g)} D^{(g)})^T = \mathbf{B}^{(g)} (\mathbf{D}^{(g)})^2 (\mathbf{B}^{(g)})^T$

$$E \|\mathcal{N}(0, \mathbf{I})\| = \sqrt{n} \Gamma\left(\frac{n+1}{2}\right) / \Gamma\left(\frac{n}{2}\right) \approx \sqrt{n} \left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$$

A função de *Heaviside* (Equação 2.2) é utilizada para parar o crescimento do caminho evolutivo da matriz de covariância (\mathbf{p}_c), quando $\|\mathbf{p}_\sigma\|$ se torna muito grande. É uma correção usada para quando o algoritmo é inicializado com o tamanho do passo pequeno.

$$h_\sigma^{(g+1)} = \begin{cases} 1 & \text{Se } \frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\sqrt{1-(1-c_\sigma)^2}} < \left(1.4 + \frac{2}{n+1} E \|\mathcal{N}(0, \mathbf{I})\|\right) \\ 0 & \text{senão} \end{cases} \quad (2.2)$$

Os parâmetros c_σ , c_c , c_1 e c_μ são as taxas de aprendizado do tamanho do passo, matriz de covariância, média e população, respectivamente. Assim como esses, os parâmetros λ , μ e $w_{i=1, \dots, \mu}$ são configurados com valores padrão que servem para um espectro grande de problemas. Em Hansen (2005) é indicado que não se deve alterá-los a não ser que o problema tenha alguma característica muito específica que requer a necessidade de alterar essas configurações. No entanto, a configuração do λ (tamanho da população) é alterada, principalmente para tratar problemas com muitos parâmetros que superfícies de busca complexas e cheias de ótimos locais, como a função Rastrigin. O efeito da decomposição em autovalores (\mathbf{D}) e autovetores (\mathbf{B}) pode ser visualizado na Figura 4.

2.2.2 Funções Mal-Condicionadas e não-separáveis

O comportamento do Enxame de Partículas (*Particle Swarm Optimization* - PSO) (KENNEDY, 2010) e do CMA-ES em funções objetivo mal-condicionadas e não-separáveis foi investigado em Hansen et al. (2011). Uma função objetivo é separável quando é possível decompô-la em problemas menores, por exemplo, a obtenção do valor ótimo para i -ésima coordenada não depende das outras coordenadas. No caso ótimo uma função separável com D dimensões poderia ser resolvida otimizando D problemas unidimensionais. Problemas separáveis são mais fáceis de serem resolvidos com uma

¹*eigendecomposition*

Algoritmo 1 CMA-ES

Entrada Defina os parâmetros λ , μ , $w_{i=1,\dots,\mu}$, c_σ , c_c , c_1 e c_μ com seus valores padrão de acordo com Hansen (2005) ou Loshchilov (2013).

- 1:
- 2: ▷ O indicador da geração g foi ocultado das variáveis para melhorar a clareza das equações.
- 3: **Inicializa:**
- 4: Definir $\mathbf{p}_c = 0$, $\mathbf{p}_\sigma = 0$, matriz de covariância $\mathbf{C} = \mathbf{I}$, e $g = 0$.
- 5: Escolha o centroid inicial da distribuição $\mathbf{m} \in \mathbb{R}^D$ e o tamanho do passo $\sigma \in \mathbb{R}_+$ de acordo com o problema.
- 6:
- 7:
- 8: **Enquanto** critério de término não for atingido **faça**
- 9: Amostra uma nova população, para $i = 1, \dots, \lambda$

$$\begin{aligned} \mathbf{z}_i &\sim \mathcal{N}(0, \mathbf{I}) \\ \mathbf{y}_i &= \mathbf{B}\mathbf{D}\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{C}) \\ \mathbf{x}_i &= \mathbf{m} + \sigma \mathbf{y}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{C}) \end{aligned} \quad (2.3)$$

- 10:
- 11: Seleção
- 12: $\langle \mathbf{y} \rangle_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$, onde $\sum_{i=1}^{\mu} w_i = 1, w_i > 0$
- 13:

14: Recombinação

$$15: \quad \mathbf{m} \leftarrow \mathbf{m} + \sigma \langle \mathbf{y} \rangle_w = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$$

- 16:
- 17: Adaptação da matriz de covariância

$$18: \quad \mathbf{p}_c = (1 - c_c) \mathbf{p}_c + h_\sigma \sqrt{c_c(2 - c_c) \mu_{eff} \langle \mathbf{y} \rangle_w} \quad (2.4)$$

$$19: \quad \mathbf{C} = (1 - c_1 - c_\mu) \mathbf{C} + c_1 (\mathbf{p}_c \mathbf{p}_c^T + \delta(h_\sigma) \mathbf{C}) + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T \quad (2.5)$$

- 20:
- 21: Adaptação do tamanho do passo

$$22: \quad \mathbf{p}_\sigma = (1 - c_\sigma) \mathbf{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma) \mu_{eff} \mathbf{C} \langle \mathbf{y} \rangle_w} \quad (2.6)$$

$$23: \quad \sigma = \sigma \times \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{E[\|N(0, \mathbf{I})\|]} \right) \right) \quad (2.7)$$

- 24:
- 25: Incrementa geração
- 26: $g \leftarrow g + 1$

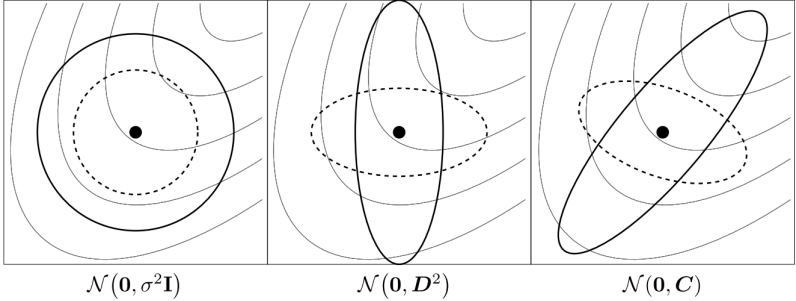


Figura 4 – Exemplo de como a Equação 2.3 afeta a matriz de covariância. Esquerda: $z^{(g+1)}$. Meio: $D^{(g)}z^{(g+1)}$. Direita: $B^{(g)}D^{(g)}z^{(g+1)}$. Figura retirada de (HANSEN, 2005)

s

complexidade linear com relação ao tamanho do problema, enquanto problemas não-separáveis são difíceis de resolver e envolvem um crescimento exponencial da complexidade no número de dimensões.

Já funções mal-condicionadas correspondem a situações em que pequenas diferenças nas variáveis podem levar a resultados muito diferentes na função objetivo, em várias ordens de grandeza. Funções objetivo mal-condicionadas dificilmente conseguem guiar a busca para soluções ótimas, sendo um desafio para qualquer método de otimização (HANSEN et al., 2011).

A Tabela 1 faz um paralelo entre o PSO e o CMA-ES, segundo as conclusões obtidas por Hansen et al. (2011). O PSO é um algoritmo base para vários métodos mais atuais (FIELDSEND, 2014; FLETCHER, 2013; ZHANG; ZHANG; LI, 2005), por isso a comparação pode ser estendida a outros algoritmos que herdaram suas propriedades.

O CMA-ES apresenta um algoritmo mais robusto. Os resultados mostram que explorar as correlações entre coordenadas através da matriz de covariância melhora a convergência do método em problemas mal-condicionados e não-separáveis, superando o PSO por um fator de $\kappa(\mathbf{A})^{\frac{3}{4}}$, onde $\kappa(\mathbf{A})$ retorna o número de condicionamento:

$$\mathbf{A} = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\|$$

na qual \mathbf{A} é uma matriz de covariância. Para um número de condicionamento moderado, e.g. 10^4 , $\kappa(\mathbf{A})$ pode chegar a alguns milhares. Sendo uma vantagem muito maior do que o fator de três com o qual o PSO superou o CMA-ES em

Tabela 1 – Comparação entre PSO e CMA-ES sob as questões de invariância do método e mal-condicionamento do problema.

Propriedade	PSO	CMA-ES
Bom desempenho em funções separáveis	X	X
Bom desempenho em funções não-separáveis		X
Invariante à transformações que preservam a ordem	X	X
Invariante ao dimensionamento do espaço de busca	X	X
Invariante ao dimensionamento de uma variável	X	X
Invariante sob rotações do espaço de busca		X
Invariante à transformações lineares no espaço de busca		X
Implementação simples	X	

problemas separáveis (HANSEN et al., 2011).

2.3 OTIMIZAÇÃO MULTIMODAL

Os Algoritmos Evolucionários lidam naturalmente com múltiplas soluções por utilizarem o conceito de população para explorar o espaço de busca. No entanto, esse conceito é explorado de maneira que a sua utilização, na forma como é empregado nas suas versões canônicas, não é viável (DAS et al., 2011). Tal conclusão foi alcançada por Mahfoud (1994) que estudou os efeitos da *oscilação genética* (*Genetic drift*). O uso de operadores de seleção e recombinação que privilegiam os indivíduos mais aptos, produz um segundo efeito conhecido como *pressão evolutiva*. Esse efeito direciona todos os indivíduos da população para o mesmo lugar, causando uma queda na diversidade da população. Exatamente o contrário do que se espera de uma otimização multimodal.

Os métodos de *niching* (MAHFOUD, 1995a) são técnicas projetadas para realizar a manutenção da diversidade da população em algoritmos evolucionários ou baseados em enxame (e.g. Epitropakis, Li e Burke (2013) e Fieldsend (2014) respectivamente), de forma que múltiplas soluções possam ser encontradas.

2.4 NICHING

A característica de *niching*², é uma característica que propicia a busca e manutenção de múltiplas regiões favoráveis do espaço de busca. Essas regiões normalmente contêm uma solução ótima, prevenindo o algoritmo de convergir para apenas uma solução. Tais áreas de atração são chamadas de *nichos* (MAHFOUD, 1995a).

Os nichos podem ser encontrados em todos os ecossistemas, e representam uma quantidade finita de recursos que pode ser obtida através de alguma característica e que é compartilhada pelos indivíduos que a possuem. Esses micro-ecossistemas permitem a formação e manutenção de diferentes características, uma vez que essa divisão ocorre por algum motivo, uma barreira ou distância por exemplo, a probabilidade de ocorrer um cruzamento entre indivíduos de grupos diferentes é menor do que a probabilidade de ocorrer um cruzamento dentro do subgrupo, e portanto, reduzindo a oscilação genética (MAHFOUD, 1994; PREUSS; SCHÖNEMANN; EMMERICH, 2005).

Para reproduzir esse comportamento em um método de otimização multimodal, a grosso modo, deve selecionar os melhores indivíduos ind_i , com $0 < i \leq NC$, definir um raio r , considerar que o recurso disponível de cada nicho é o valor de aptidão de cada indivíduo selecionado e portanto teríamos NC nichos. Em seguida, elimine os nichos que se sobrepõem utilizando c_i como centro e o raio r . Por fim, defina uma probabilidade de cruzamento maior dentro de cada nicho do que entre nichos. Isso implica que indivíduos mais semelhantes ao c_i , terão uma probabilidade maior de continuarem dentro do i -ésimo nicho.

Os métodos de *niching* podem ser classificados como: sequenciais e paralelos. Os métodos paralelos estão fortemente relacionados aos métodos de otimização baseados em população, como os algoritmos evolucionários. Os métodos sequenciais executam um nicho a cada momento, e podem utilizar a informação da execução anterior para guiar a próxima. Apesar de existirem bons métodos de *niching* sequencial, os métodos mais atuais estão sendo desenvolvidos usando o conceito de *niching* devido as possibilidades de paralelismo. No entanto a seção a seguir faz uma breve discussão sobre os métodos sequenciais.

²Termo em inglês que pode ser traduzido como: "formação de nichos".

2.4.1 *Niching* Sequencial

Beasley, Bull e Martin (1993) demonstraram uma técnica que penalizava o valor de aptidão das soluções que se encontravam próximas a uma solução ótima já encontrada. Uma solução é considerada próxima quando ela está dentro do raio de cobertura do nicho. Neste caso, o nicho era representado pela própria solução encontrada. A busca era realizada por um Algoritmo Genético (AG), que era reinicializado sempre que estagnava, armazenando a última solução encontrada em cada execução. Ando, Sakuma e Kobayashi (2005) utilizaram um modelo de isolamento adaptativo aonde foi aplicada uma técnica de agrupamento durante a execução do AG para isolar subpopulações. Zhang, Zhang e Li (2005) utilizaram o PSO sequencialmente em conjunto com uma modificação na função de aptidão para isolar os atratores e buscá-los de forma independente. Embora os métodos de *niching* sequencial (*Sequential Niching* - SN) terem demonstrado resultados positivos em *benchmarks* de otimização multimodal, Mahfoud (1995b) comparou os métodos de *niching* utilizando uma combinação de AG com técnicas de desclassificação como exemplo de *niching* sequencial. Seus experimentos demonstraram que os métodos paralelos (*Parallel Niching* - PN) tinham um desempenho melhor tanto em termos de velocidade como número de ótimos encontrados. Ele concluiu que os métodos paralelos superam os métodos sequenciais pelas seguintes razões: desclassificação é incapaz de eliminar completamente o problema de localizar as mesmas soluções repetidamente, e ainda, ela pode realocar ou destruir um ótimo de interesse ou mesmo criar falsos atratores. Tais aspectos dificultam a localização dos locais ótimos remanescentes, principalmente em problemas com superfícies de erro muito “acidentadas”. Caso a função de aptidão seja simples, pelo fato de ser um método caixa-preta que desconhece a complexidade da função, ele pode torná-la mais complicada. Essas adversidades contribuem para o caminho dos métodos de *niching* paralelos.

2.5 NICHING PARALELO

De acordo com Mahfoud, “a diversidade é um termo genérico que descreve a variação ou falta de similaridade entre os objetos de uma coleção” (MAHFOUD, 1995a). É intuitivo imaginar um problema multimodal sendo resolvido por uma população de indivíduos que evoluem separadamente com o objetivo de serem ótimos em “nichos” diferenciados. No entanto, é preciso definir uma métrica de similaridade que possa ser usada para verificar a diversidade da população como, por exemplo, a soma das distâncias euclidianas entre os indivíduos da população.

Geralmente os algoritmos de otimização inicializam sua população utilizando uma amostragem uniforme sobre o espaço de busca, o que garante a diversidade máxima da população. Conforme as gerações passam, essa diversidade irá diminuir. A ideia é que a diversidade diminua até servir apenas para manter as soluções encontradas. Mahfoud argumenta que "... a diversidade é útil se ajudar a alcançar algum objetivo" (MAHFOUD, 1995a). Significando que o objetivo do método de *niching* não é manter a diversidade alta, mas sim, uma diversidade "útil".

2.5.1 Técnicas Clássicas

Nesta seção serão apresentados alguns algoritmos de *niching* paralelo que foram desenvolvidos nos primórdios dos estudos de otimização multimodal, porém que continuam servindo de inspiração e base para os métodos atuais.

O *Crowding* é uma abordagem que restringe a competição apenas entre membros da população que são similares, i.e., entre os indivíduos que pertencem ao mesmo nicho. Cavicchio (1970) apresenta um esquema de pré-seleção, aonde os indivíduos da população candidata substituem seus "pais" se tiverem uma aptidão maior que a deles. A razão para este comportamento se deve a similaridade entre soluções "pai" e "filho" que teoricamente deve ser alta e portanto apenas um necessita ser mantido. A grande vantagem dessa abordagem é o fato de não adicionar nenhuma complexidade ao algoritmo original, além de não necessitar de nenhum parâmetro de controle ou mesmo uma medida de similaridade. O conceito de *crowding* (JONG, 1975), foi desenvolvido como um meio de manter a diversidade prevenindo a convergência. O método comparava as soluções candidatas com uma amostra aleatória da população e removia o indivíduo mais similar da amostra.

Outra técnica bastante conhecido é o *Sharing*. Holland (1975) introduziu o conceito de que os "recursos" provenientes de um nicho devem ser compartilhados entre os indivíduos que se "especializaram" no mesmo. Basicamente os indivíduos devem compartilhar o valor de aptidão obtido naquele nicho (região de atração), conseqüentemente afastando as soluções de nichos populosos, e evitando que áreas que já são bem conhecidas continuem sendo exploradas. A ideia é que esse processo se estabilize, mantendo os nichos com um número de indivíduos proporcional a qualidade daquela região. Na prática, os métodos modificam o valor de aptidão dos indivíduos de acordo com a proximidade dos membros da população.

O valor de aptidão compartilhado f^s de um indivíduo \vec{x}_i é calculado dividindo o próprio valor de aptidão pela soma dos valores proporcionais a

distância em que este indivíduo está dos outros,

$$f_s(\vec{x}_i) = \frac{f(\vec{x}_i)}{\sum sh(d_{i,j})} \quad (2.8)$$

aonde $sh(d_{i,j})$ é a função de partilha e $d(i, j)$ é a distância entre os indivíduos \vec{x}_i e \vec{x}_j . A distância utilizada normalmente é a distância euclidiana, porém outra métrica apropriada ao problema pode ser utilizada. Goldberg e Richardson (1987) definem a função de partilha $sh(d_{i,j})$ como,

$$sh(d_{i,j}) = \begin{cases} 1 - \left(\frac{d_{i,j}}{\sigma_{rad}}\right)^\alpha, & \text{if } d_{i,j} < \sigma_{rad} \\ 0, & \text{otherwise} \end{cases} \quad (2.9)$$

aonde σ_{rad} é o raio do nicho. Como na maioria dos algoritmos de *niching*, definir o valor desse parâmetro é crucial para obter um bom desempenho do algoritmo. Normalmente é necessária alguma informação *a priori* sobre o problema para estimar corretamente o raio. Outro ponto desfavorável é que este método requer, para cada indivíduo, $O(NP)$ avaliações de similaridade para atualizar o valor de aptidão de toda uma população de tamanho NP .

O *Clearing* (PETROWSKI, 1996) é uma das técnicas de *niching* mais simples de ser aplicadas, sendo similar ao conceito de *Sharing*. O *Clearing* também define um parâmetro de raio (*clearing radius*). Entretanto, este raio define a área na qual apenas k indivíduos (aqueles com maior aptidão) irão “sobreviver”, removendo o restante. Ao contrário do *Sharing*, o *Clearing* dá todo o recurso disponível para o indivíduo vencedor, no caso mais utilizado de $k = 1$. Um das vantagens é o fato do *clearing* reduzir o número de indivíduos necessários para manter um nicho, reduzindo a complexidade do método.

2.5.2 Técnicas de *Niching* explícito

Algumas técnicas de *niching* dividem explicitamente a população em subgrupos, onde cada grupo é otimizado independentemente. Entretanto, isso não impede que exista algum nível de comunicação entre as subpopulações. Pois caso não ocorra nenhuma comunicação entre elas, o método seria equivalente a um método de SN. Para ser considerado realmente explícito, o método de *niching* não pode possuir uma fase de exploração global, dado que algoritmo nunca “enxerga” a sua população como um todo. Cada subpopulação conhece apenas a informação presente nos indivíduos a sua disposição. A implicação desta abordagem é uma busca concentrada em uma regiões diferentes do espaço de busca, facilitando principalmente a paralelização do algoritmo, mas

resultando em uma busca exploratória (global) menos eficaz que os métodos implícitos (DAS et al., 2011). A seguir, dois conceitos importantes para trocar informações entre os nichos no modelo explícito.

A *Restrição de Cruzamento (Mating Restriction)* surgiu como um conceito para melhorar o algoritmo de *fitness sharing*. A motivação para restringir a combinação entre indivíduos de populações diferentes é evitar gerar soluções que movam o nicho de lugar. Entretanto, restringir a combinação não é o suficiente para manter a diversidade da população, como observado por Mahfoud (1995a). Facilmente combinada com a técnica de *clustering*, ela restringe a combinação de indivíduos de diferentes nichos. Várias abordagens implícitas foram modificadas para suportar esse tipo de restrição. Na verdade, as restrições de combinação são parte da definição de um método explícito, e portanto quando um método implícito aplica restrições de combinação entre nichos ele passa a ser explícito.

Shir e Bäck (2006) adotaram a ideia de nichos dinâmicos para seu algoritmo. Basicamente cada subpopulação gerava uma quantidade fixa de indivíduos por geração. Posteriormente o método foi aprimorado para ser usado em conjunto com o CMA-ES, para adaptar o tamanho do raio (SHIR; EMMERICH; BÄCK, 2010). No método proposto, esse conceito é empregado para evitar troca de informação excessiva entre os nichos.

Seguindo a mesma ideia, a utilização de *Modelos de Ilha (Island Models)* (MARTIN; LIENIG; COHOON, 2000) pode ser vista como um caso especial da Restrição de Cruzamento. Esses modelos são baseado na teoria evolutiva que propõe que a maior parte das populações de organismos de reprodução sexuada vivenciam poucas mudanças ao longo de sua história, e quando estas finalmente ocorrem, são eventos pontuais e rápidos. Portanto cada subpopulação é separada em “ilhas” isoladas. Após um número pré-determinado de gerações, ocorre uma troca de informação entre as “ilhas” através de migração. Diversas variantes desse modelo foram desenvolvidas para atender características específicas. Basicamente as variações se restringem a modificar a forma como a migração ocorre (POTTER; JONG, 2000) e a estrutura topológica das “ilhas” (PETROWSKI; GENET, 1999).

Um exemplo é o algoritmo evolucionário baseado em multi-nações proposto por Ursem (1999), divide a população em subpopulações (nações) que são evoluídas separadamente. O governo é um subconjunto da população, formado pelos indivíduos mais aptos. Uma política é um único ponto calculado como a média do governo. Basicamente, o algoritmo avalia se um indivíduo ainda pertence a nação ou se ele migrou para outra nação utilizando uma técnica chamada de *Hill-Valley (Hill-Valley - HV)*, que amostra pontos na reta que liga dois indivíduo, testando entre o indivíduo e a política. Outro teste serve para verificar se duas nações devem ser combinadas, aplicando o HV

entre duas políticas.

O principal problema desse método é que o número de avaliações da função objetivo aumenta drasticamente, já que o HV é aplicado para cada indivíduo, incluindo as políticas, sendo necessários vários pontos para que as decisões sejam precisas (e.g. combinar ou não combinar). Utilizando uma ideia similar o método de Streichert et al. (2004) utiliza um agrupamento baseado em densidade ao invés do HV, reduzindo o número de avaliações da função objetivo.

Recentemente, o método proposto por (FIELDSEND, 2014) utiliza o HV para decidir quando se deve combinar e quando se deve criar nichos. O NMESIS também utiliza o HV com o mesmo objetivo. O NMESIS empresta alguns conceitos do modelo de ilha para definir o processo de controle de sobreposições.

2.6 TRABALHOS RELACIONADOS

O NEA2 foi proposto por Preuss (2012), e até o início dessa dissertação, era o método declarado vencedor da competição organizada pelo Congresso de Computação Evolucionária (CEC 2013). O NEA2 é uma variação sequencial do NEA (PREUSS, 2010) que é um *niching* paralelo. A implementação do NEA2 é simples e depende basicamente do algoritmo de inicialização que utiliza o agrupamento por “melhor-mais-próximo” (*Nearest Better Clustering - NBC*) com uma nova regra de corte. O NEA2 não utiliza nenhuma informação extra gerada pelo CMA-ES para acelerar ou melhorar os resultados.

O *Niched CMA-ES* de Shir, Emmerich e Bäck (2010) utiliza elementos adaptativos do CMA-ES para contornar o problema do *raio de abrangência do nicho*, adaptando o valor do raio ou usando a distância de *Mahalanobis* para detectar se dois nichos estão muito próximos. Basicamente, cada nicho era representado por uma instância do CMA-ES e a população era dividida em dois grupos que a cada iteração eram reagrupados em novos nichos. Um grupo representa os centroides enquanto o outro representa os indivíduos gerados por cada nicho, i.e. todas as populações.

O GCES (DONG; ZHOU, 2014) (*Gaussian Classifier Evolution Strategy - GCES*) utiliza uma abordagem bayesiana para classificar os indivíduos para seus respectivos nichos, demonstrando que seu método é análogo a um *perceptron* com funções de transformação gaussianas (DONG; ZHOU, 2014). Além disso, algumas alterações no algoritmo original do CMA-ES foram feitas.

O GCES e o *Niched CMA-ES* são os métodos que utilizam melhor o conhecimento armazenado pelo CMA-ES. Apesar disso, ambos não supe-

ram o desempenho do NEA2, principalmente com respeito a estabilidade dos resultados. O GCES conseguiu resultados bons em funções complexas mas teve problemas em função triviais. Mesmo nas funções complexas seus resultados foram melhores apenas quando o requisito de precisão era baixo ($1e^{-1}$). Alterar o fluxo normal do CMA-ES inserindo novos indivíduos ou realocando o centroide pode causar a degeneração da matriz de covariância (HANSEN, 2005).

O NMMSO (FIELDSEND, 2014), utiliza o PSO como algoritmo base e para detectar sobreposição de nichos ele aplica o *Hill-Valley*. No entanto, ele realiza esse procedimento a cada iteração e verifica todos os pares de nichos independente da distância entre eles e utiliza apenas o ponto médio. Se caso esse ponto não for um vale, mesmo que os nichos estejam distantes, eles serão combinados. Isso pode gerar efeitos indesejados como eliminar nichos promissores.

Como foi observado que poucos métodos se importavam com a inicialização da população, e que o NEA2 possui um mecanismo de inicialização poderoso mas pouco explorado, essa área fez parte da pesquisa desta dissertação. O resultado culminou na publicação de um artigo no Congresso de Computação Evolucionária (*IEEE Congress of Evolutionary Computing - CEC*) na edição de 2014. O TN-CMA-ES (PEREIRA; NETO; ROISENBERG, 2014), como foi chamado devido ao uso de técnicas de topologia e *niching*, propôs uma versão estendida do NBC utilizando o *Hill-Valley* (URSEM, 1999).

2.7 DISCUSSÃO

Este capítulo reuniu os conceitos de otimização e *niching* para embasar as escolhas feitas nessa dissertação. Ao escolher os algoritmos evolucionários por sua flexibilidade, robustez e estado-da-arte não há melhores opções além do CMA-ES (HANSEN; KERN, 2004; HANSEN, 2009; LOSHCHILOV, 2013). As propriedades do CMA-ES como a invariância com relação à transformações lineares da função objetivo e adaptabilidade são raras de se encontrar em outros métodos, principalmente com o rigor matemático do CMA-ES (HANSEN, 2005). Herdar essas propriedades foi a principal motivação para utilizar o CMA-ES como algoritmo base para o NMESIS.

Quanto a abordagem de *niching*, apesar dos métodos sequenciais serem mais fáceis de implementar e terem um ótimo representante, o NEA2, as possibilidades permitidas pelos métodos de *niching* paralelos são mais interessantes, uma vez que o método pode ser paralelizado, aplicado em problemas dinâmicos e também em problemas de tempo-real. E com os recentes resultados obtidos pelo NMMSO (FIELDSEND, 2014) e GCES (DONG; ZHOU, 2014),

os métodos paralelos também mostram ótimos resultados. Isso motivou a definição do NMESIS como um método de *niching* paralelo. Ser explícito foi uma necessidade imposta pela utilização do CMA-ES, pois os indivíduos da sua população não dependem apenas do próprio estado atual como ocorre no GA ou PSO e portanto não podem ser tratados individualmente. Apenas a instância do CMA-ES associada ao nicho é quem altera a população.

O próximo capítulo foca na proposta deste trabalho, utilizando a combinação do CMA-ES com *niching* paralelo semelhante aos trabalhos de Shir, Emmerich e Bäck (2010) e Dong e Zhou (2014), porém com controle diferenciado de comunicação entre nichos e com uma estratégia de exploração mais avançada.

3 PROPOSTA

No Capítulo 2 foram apresentados conceitos e métodos usados como base para construção de técnicas de otimização multimodal (OM). Compreender o relacionamento entre esses conceitos e as técnicas já existentes é o primeiro passo para desenvolver um método de OM efetivo. No Capítulo 1, temos a opinião de Fieldsend (2014) sobre as características que um método de otimização efetivo deve possuir. São elas:

1. Ser dinâmico com relação ao número de soluções que devem ser encontradas;
2. Adaptar-se automaticamente;
3. Possuir um mecanismo de busca local;

Em seu método, o autor buscou evidenciar essas características, assim como também argumentou sobre a presença delas no atual vencedor¹ do *benchmark* CEC 2013, o NEA2 (PREUSS, 2012). Ambos os métodos apresentaram ótimos resultados no *benchmark*, no entanto quando são analisados apenas os resultados dos problemas com número de parâmetros maior que cinco, ainda há muito o que ser melhorado. Esse trabalho propõe uma combinação de algoritmos voltados para otimização multimodal com o objetivo principal de obter melhores resultados em problemas multimodais, não-separáveis e mal-condicionados, de tamanho médio (5-20 dimensões).

Dentre as opções de algoritmos de otimização disponíveis, a Estratégia de Evolução com Covariância Adaptativa (ou CMA-ES - *Covariance Matrix Adaptation Evolution Strategy*) foi escolhida. Esse método foi descrito em detalhes na Subseção 2.2.1. O CMA-ES é um método de otimização recente que vem sendo utilizado e estudado. Apesar da sua complexidade teórica, ele expõe todo o mecanismo de busca e adaptação, permitindo extrair informações importantes sobre o problema e sobre o desempenho do método durante sua execução (HANSEN et al., 2011; LOSHCHILOV et al., 2014; LOSHCHILOV, 2014). Sua capacidade de lidar com problemas complexos, não-separáveis e mal-condicionados, além de ser invariante a uma série de transformações, permite que esse método seja utilizado em diversos tipos de problemas de otimização (HANSEN, 2005).

No entanto, o CMA-ES não foi desenvolvido com o objetivo de obter múltiplas soluções, o que significa que executar o CMA-ES para um problema

¹Não há referência específica para o resultado da competição, porém o documento está disponível em <http://goanna.cs.rmit.edu.au/xiaodong/cec13-niching/competition/Niching-final-presentation-CEC2013.pdf>

multimodal, retornaria apenas uma única solução global, ou pior, ficaria preso em um ótimo local. Alguns dos métodos de *niching* apresentados na Subseção 2.6 utilizam o CMA-ES, como por exemplo o NEA2. Este aplica o CMA-ES para explorar as soluções sequencialmente, reiniciando-o sempre que uma solução é encontrada ou quando o algoritmo diverge. Já o método de *Niched CMA-ES* (SHIR; EMMERICH; BÄCK, 2010) aplica uma técnica de *niching* paralelo, em que múltiplas buscas usando o CMA-ES são executadas simultaneamente. Um dos principais obstáculos para desenvolver um algoritmo de *niching* que possa ser usado na prática, é justamente evitar que essas buscas, paralelas ou sequenciais, encontrem as mesmas soluções repetidamente. Existem diversas maneiras de evitar essas situações, entretanto existe um custo associado a essa verificação de redundância da solução que deve ser levado em conta na construção do método. Por exemplo, o NEA2 apenas tenta inicializar as buscas em pontos distintos, mas não possui nenhum mecanismo de verificação entre as iterações. Já o *Niched CMA-ES* inicializa sua população uniformemente e utiliza um parâmetro adaptativo para definir qual é a área ocupada pelo nicho e eliminar aqueles que estão ocupando o mesmo local.

Uma das contribuições desta dissertação é conciliar essas duas formas de atuar sobre o problema, através de uma inicialização que permita que o método comece bem posicionado e de verificações durante a sua execução. O método proposto é um *niching* paralelo e explícito, ou seja, as buscas serão executadas concorrentemente e a divisão dos nichos é feita explicitamente. A comunicação entre os nichos é feita utilizando dois operadores, um para detectar redundâncias e outro para explorar globalmente o problema. Essas modificações deram origem a proposta de uma Estratégia de Evolução Multimodal baseada em Multi-população (*Niched Multi-population Evolution Strategy with Improved Search - NMESIS*). Nos capítulos a seguir, o método proposto neste trabalho será referenciado como NMESIS.

A principal vantagem de desenvolver um método de *niching* explícito é que automaticamente o número de verificações necessárias (distância ou qualquer técnica de similaridade) é reduzido de $O(\Lambda)$ para $O(\Phi)$, onde Λ é a soma de cada subpopulação (nicho), Φ é o número de nichos e $\Lambda \geq \Phi$. Ou seja, se a formação de nichos for eficiente, haverá um ganho considerável. Essa redução ocorre, pois como a informação que entra e sai do nicho é controlada, a verificação só precisa ser feita entre os nichos, preferencialmente, com relação ao centroide dele. A Seção a seguir trata de como a representação da população do CMA-ES é utilizada para definir o centroide e a população do nicho.

3.1 MODELAGEM DOS NICHOS VIA MISTURAS GAUSSIANAS

Para explorar o conceito de localidade, foi necessário implantar um mecanismo que permite restringir a busca a uma região bem definida. Como o CMA-ES representa a sua população como uma distribuição gaussiana, ao controlar o desvio padrão dessa distribuição é possível controlar a área de amostragem e portanto controlar a busca. O NMESIS é um método de *niching* que utiliza múltiplas populações, e cada população é representada por uma gaussiana multivariada (\mathcal{N}^D), o resultado de cada iteração do algoritmo é um Modelo de Mistura de Gaussianas (*Gaussian Mixture Model* - GMM) (REYNOLDS, 2009). Essa modelagem permite o uso de técnicas já desenvolvidas para esse tipo de modelo como regras de decisão Bayesiana e otimização (RUNNALLS, 2007). Na Figura 5, um GMM gerado a partir de dois nichos, que aparentam estar convergindo na mesma direção.

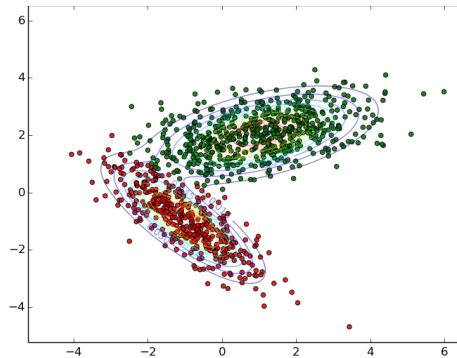


Figura 5 – Dois nichos representados por suas respectivas distribuições (\mathcal{N}^D).

Através dessa modelagem é possível tratar o problema de otimização multimodal, como o equilíbrio entre minimizar o número de gaussianas ao mesmo tempo que maximiza-se a probabilidade de obter melhores soluções. Uma vez que a probabilidade está relacionada à aptidão da solução, i.e. soluções mais aptas possuem maior probabilidade de serem escolhidas. Essa visão também é reforçada pelo modelo apresentada na Figura 6. O modelo biogeográfico apresentado está diretamente ligado aos métodos de *niching*, principalmente aqueles que usam o modelo de ilhas para definir os nichos. Entretanto, qualquer método de *niching* precisa encontrar um equilíbrio entre criar e eliminar nichos. Alguns métodos utilizam o conceito de migração, em que indivíduos (e.g. soluções) migram de um nicho para outro, em determina-

das condições novos nichos podem se formar através da migração. Já outros nichos podem perder todos os indivíduos, reduzindo o número de nichos. Essa região de equilíbrio é sinalizada pelo retângulo tracejado na Figura 6.

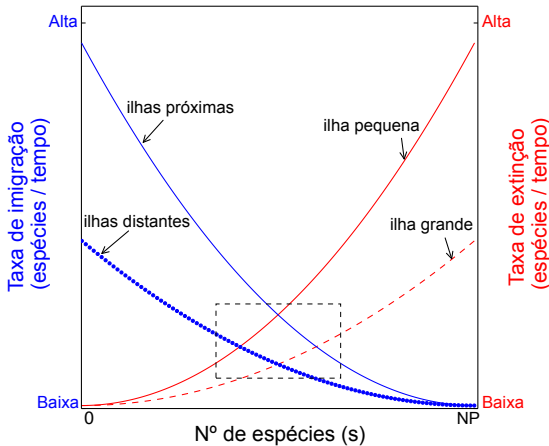


Figura 6 – Modelo biogeográfico de ilhas (MACARTHUR, 1967).

3.1.1 Função de Similaridade

Definir uma métrica de similaridade entre os nichos é importante para podermos detectar quando dois nichos estão ocupando a mesma região, que em outras palavras significa que o método está produzindo pelo menos uma busca redundante. Como o conjunto de nichos em execução é tratado pelo NMESIS como uma GMM, é coerente usar uma métrica que seja adequada a essa representação.

A *Divergência de Kullback-Leibler* ($d_{KL}(A||B)$) (KULLBACK; LEIBLER, 1951) é uma medida utilizada para comparar duas distribuições. Essa medida leva em conta diversas características da distribuição, como a matriz de covariância, sendo mais confiável do que apenas calcular a distância euclidiana entre os centroides. No entanto, a $d_{KL}(A||B)$ é um cálculo custoso e como todos os nichos são assumidos como distribuições gaussianas é possível utilizar uma simplificação, através da distância de Mahalanobis, $||\vec{x}_k - \vec{y}||_{\Sigma_k}^2$, onde Σ_k é a matriz de covariância da k -ésima distribuição.

Por exemplo, dadas duas distribuições, A e B , com os respectivos

centroides c_A e c_B , é feito

$$md_{AB} = \|\vec{c}_A - \vec{c}_B\|_{\Sigma_A}^2. \quad (3.1)$$

$$md_{BA} = \|\vec{c}_B - \vec{c}_A\|_{\Sigma_B}^2. \quad (3.2)$$

O calculo da distância é feito nas duas direções porque a distância de Mahalanobis é assimétrica pois leva em consideração apenas a matriz de covariância de uma das distribuições por vez. Com essas informações, a métrica de similaridade fica mais robusta e correta. A utilização da distância de Mahalanobis é amparada pela Regra de Decisão Bayesiana (RDB), que é usada neste trabalho como mecanismo de classificação das soluções (LIU, 2007). De acordo com a RDB, \vec{x} pertence à componente ω_k se:

$$P(\vec{x}|\omega_k) \cdot P(\omega_k) \leq P(\vec{x}|\omega_j) \cdot P(\omega_j), \quad (3.3)$$

pode-se então obter um classificador gaussiano, definido como:

$$g_k(\vec{x}) = \log(p(\vec{x}|\omega_k)) \cdot \log(p(\omega_k)). \quad (3.4)$$

O modelo Gaussiano da Equação 3.4, pode ser rescrito como

$$g_k(\vec{x}) = -\frac{1}{2}(\vec{x} - \vec{m}_k)^T \Sigma_k^{-1} (\vec{x} - \vec{m}_k) - \frac{n}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_k| + \log P(\omega_k), \quad (3.5)$$

aonde n é o número de dimensões, Σ_k é a matriz de covariância do componente (nicho) e \vec{m}_k é o centroide. Com $g_k(\vec{x})$ definida é possível elaborar uma classificação gaussiana:

$$k^*(\vec{x}) = \arg \max_k g_k(\vec{x}), \quad (3.6)$$

qual o nicho k^* com maior probabilidade de conter a solução \vec{x} . Pode-se alterar a otimização para ficar mais intuitiva com a ideia de nichos e proximidade. A Equação 3.6 pode ser reescrita como a seguinte minimização:

$$k^*(\vec{x}) = \arg \min_k [\|\vec{x} - \vec{m}_k\|_{\Sigma_k}^2 + \alpha_k], \text{ aonde} \quad (3.7)$$

$$\|\vec{x} - \vec{m}_k\|_{\Sigma_k}^2 = (\vec{x} - \vec{m}_k)^T \Sigma_k^{-1} (\vec{x} - \vec{m}_k), \text{ e} \quad (3.8)$$

$$\alpha_k = n \log(2\pi) + \log |\Sigma_k| - 2 \log P(\omega_k) \quad (3.9)$$

aonde $\|\vec{x} - \vec{m}_k\|_{\Sigma_k}^2$ é a distância de Mahalanobis, a mesma utilizada por Shir,

Emmerich e Bäck (2010) para calcular o raio do nicho. Eliminando os termos constantes, tem-se $\alpha_k = \log|\Sigma_k|$.

No entanto, é preciso decidir quando essas distâncias indicam que ambas as distribuições estão representando o mesmo nicho. Para que a distância $\|\bar{x}_k - \bar{y}\|_{\Sigma_k}^2$ seja utilizada, é preciso primeiramente definir um limiar θ , que irá decidir se o indivíduo será ou não classificado por algum nicho, como na Equação 3.10.

$$f_{class}(q_i) = \begin{cases} 1, & \text{Se } \|\bar{\mu}_k - \bar{q}_i\|_{\Sigma_k}^2 > \theta. \\ 0, & \text{senão.} \end{cases} \quad (3.10)$$

Dado um quantil τ , obtêm-se a distância (de Mahalanobis) máxima, θ , que um ponto amostral deve estar para que seja classificado como pertencente aquele quantil da distribuição.

Como a distribuição dos valores obtidos através da distância de Mahalanobis é uma distribuição chi-quadrado χ^2 , foi utilizada a inversa da função de probabilidade acumulada $\mathbf{F}^{-1}(\tau)$ para calcular o limiar θ . Essa transformação torna a configuração do método mais fácil, pois o parâmetro a ser configurado é adimensional e intuitivo para o usuário. Uma vez que basta decidir qual a probabilidade de pertencer à distribuição é aceitável. Por exemplo, se o usuário define $\tau = 0.65$, que é aproximadamente um desvio padrão de distância, todas os pontos que estiverem a uma distância menor que um desvio padrão, ou seja, pertencem a região que compreende 65% dos pontos amostrados. Na Figura 7 estão representados os histogramas referentes a três populações diferentes, com 5, 10 e 20 dimensões respectivamente. O limiar gerado por $\mathbf{F}^{-1}(\tau = 0.95)$ é representado pela linha que cruza o eixo y . Note que conforme a dimensão aumenta, o *theta* também aumenta como esperado, pois o domínio da função é maior. Outro ponto importante, é que a região eliminada é similar e portanto o método de corte é escalável com relação ao tamanho do problema.

Essa característica torna o método muito mais simples de ser configurado, uma vez que não ouve nenhuma necessidade de análise para decidir qual distância utilizar para cada tamanho de problema, bastando definir que qualquer ponto verificado com uma distância de Mahalanobis menor que dois desvios padrões (95%) pertence a distribuição que o verificou.

O uso da distância de Mahalanobis envolve cálculos matriciais proibitivos se considerarmos que eles seriam executados mais de uma vez por geração. Para calcular essa distância é necessário conhecer a inversa da matriz de covariância (Equação 3.7), o que significaria aumentar a complexidade do método. Por outro lado, essa distância nos trás um mapeamento direto entre a distribuição da população e o raio do nicho. Embora a distância $\|\bar{x} - m\|_{\Sigma}^2$ não esteja ligada diretamente ao CMA-ES, ela pode ser utilizada para corrigir

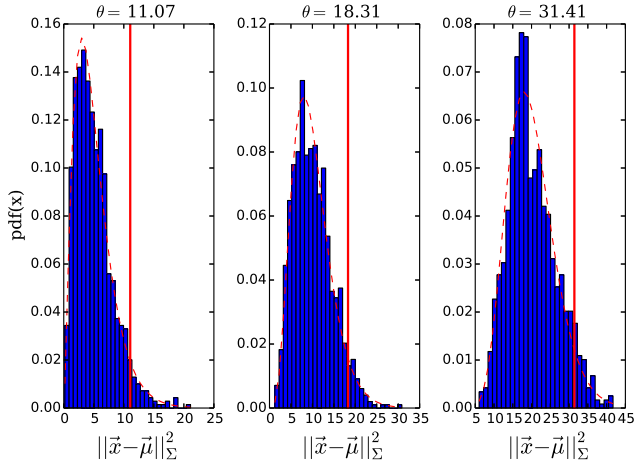


Figura 7 – Histogramas de populações com 5, 10 e 20 dimensões. A linha que cruza o eixo y mostra o θ escolhido utilizando $\tau = 0.95$ na inversa da função de probabilidade acumulada da χ^2 .

soluções e “injetá-las” na distribuição mantida pelo CMA-ES, substituindo o Σ pela matriz de covariância $C_k^{(g)}$. Desta maneira, obtêm-se uma representação do nicho mais próxima da realidade, em contraste com o modelo de raio (SHIR; EMMERICH; BÄCK, 2010; PEREIRA; NETO; ROISENBERG, 2014). Como a matriz é a mesma do CMA-ES, aproveita-se que o CMA-ES decompõe a matriz $C_k^{(g)}$ em autovalores Λ e autovetores B ,

$$C = B\Lambda B^T \quad (3.11)$$

como o CMA-ES calcula a matriz diagonal de autovalores D que é, temos

$$D = \text{diag} \left(\sqrt{\lambda_0}, \sqrt{\lambda_1}, \dots, \sqrt{\lambda_n} \right) \quad (3.12)$$

$$C = BD(DB^T) \quad (3.13)$$

após a decomposição é possível substituir o cálculo da inversa por um cálculo de multiplicação,

$$\begin{aligned}
C^{-1} &= [BD(DB^T)]^{-1} \\
&= B^{-T}D^{-T}D^{-1}B^{-1} \\
&= B \cdot \left(\frac{1}{\lambda_0}, \frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n} \right) B^T
\end{aligned} \tag{3.14}$$

Com isso o cálculo do quadrado da distância de Mahalanobis D_m^2 fica,

$$D_m^2 = (x - \mu)^T B \Lambda^{-1} B^T (x - \mu) \tag{3.15}$$

fazendo com que a projecção de $(x - \mu)$ em todos os autovetores B seja b ,

$$b = B^T (x - \mu) \tag{3.16}$$

temos,

$$\begin{aligned}
D_m^2 &= b^T \Lambda^{-1} b \\
&= \sum_i \frac{b_i^2}{\lambda_i}.
\end{aligned} \tag{3.17}$$

Utilizando a Equação 3.17 é possível obter a distância de Mahalanobis dos indivíduos da população para o centroide, e assim verificar quais indivíduos permanecem dentro do nicho.

Essa Subsecção conclui a forma como os pontos são verificados e classificados de acordo com as distribuições gaussianas existentes. A mesma técnica também é utilizada pelo NMESIS para construir o operador de migração, que detecta novos pontos dentro do raio do nicho.

3.2 ALGORITMO

As próximas subsecções descrevem os componentes do algoritmo NMESIS.

3.2.1 População Inicial

É comum entre os métodos de *niching* paralelo apenas se preocuparem em gerar uma população uniformemente distribuída sobre todo o espaço de busca. Já os métodos sequenciais tentam ser mais seletivos no momento

de escolher um novo ponto para iniciar a busca. No entanto, os métodos paralelos são métodos custosos e que iriam se beneficiar de uma inicialização que oferecesse uma população inicial que fosse menor porém mais efetiva.

Seguindo essa ideia, uma método de inicialização para problemas multimodais foi também desenvolvido durante esta pesquisa e publicado no CEC 2014 (2014 IEEE Congress on Evolutionary Computation) (PEREIRA; NETO; ROISENBERG, 2014), que utiliza informação topológica juntamente com a distância para encontrar regiões de atração com maior probabilidade de abrigarem uma solução ótima. O método foi chamado de NBC+DM.

O NBC+DM é uma extensão da técnica de inicialização utilizada pelo NEA2, que agrupa indivíduos seguindo a ordem do “melhor mais próximo”, definida pela Equação 3.18 (*Nearest Better Clustering* - NBC).

$$d_{nb}(\vec{x}, S) = \min(\{dist(\vec{x}_i, \vec{x}_j) | f(\vec{x}_j) > f(\vec{x}_i) \wedge \forall \vec{x}_j \in S\}). \quad (3.18)$$

A extensão acrescenta ao NBC uma técnica conhecida como *Hill-Valley* (URSEM, 1999), que apesar de ser uma técnica antiga, vem sendo usada em métodos novos devido à independência do cálculo de distância (FIELDSEND, 2014). O *Hill-Valley* é utilizado para verificar se dois nichos adjacentes, no grafo criado pelo NBC, estão realmente separados. Essa constatação é feita através da detecção de uma vale entre eles. O pseudocódigo apresentado no Algoritmo 2 descreve o funcionamento do NBC+DM.

Algoritmo 2 NBC+DM(P)

Entrada população P

- 1: Calcular a matriz de distância
 - 2: $G := \emptyset$
 - 3: **para todo** $\vec{x}_i \in P$ **faça**
 - 4: $\vec{x}_j := d_{nb}(\vec{x}_i, P)$
 - 5: $G \leftarrow G \cup \{aresta(\vec{x}_i, \vec{x}_j)\}$
 - 6: **para todo** aresta $e \in G$ **faça**
 - 7: Aplicar regra de corte em e
 - 8: **se** aresta e foi removida de G **então**
 - 9: $\psi \leftarrow \psi \cup Dominado(e)$ ▷ Solução deixa de ser dominada.
 - 10: **para todo** indivíduo $u \in \psi$ **faça**
 - 11: $A \leftarrow$ todas as arestas adjacentes de u
 - 12: **para todo** indivíduo adjacente $x \in A$ **faça**
 - 13: **se** u está separado de x **então**
 - 14: $\psi \leftarrow \psi \cup \{x\}$
 - 15: **retorna** ψ
-

Primeiramente, uma árvore de espalhamento que conecta todos os vértices (soluções da população P) segundo o critério da Equação 3.18 é construída. Em seguida, regras de corte são aplicadas para eliminar arestas. A regra utilizada no NMESIS é

$$f_{regra}(q_i) = \begin{cases} \text{Remove aresta,} & \text{Se } d_{nb}(\vec{x}, G) > 2\mu_{nb}(G). \\ \text{Mantêm aresta,} & \text{senão.} \end{cases} \quad (3.19)$$

onde $\mu_{nb}(G)$ é a média do tamanho de todas as arestas do conjunto G .

Cada vértice pode ter várias arestas entrantes, mas apenas uma saínte. Isso significa que um vértice pode “dominar” vários vértices, mas só pode ser “dominado” por um. Uma vez que essa aresta saínte for removida, o vértice deixa de ser dominado. Vértices não-dominados são chamados de “sementes”, pois serão utilizados como centroides dos primeiros nichos. A função *Dominado* (linha 9) retorna o vértice de menor aptidão da aresta, ou seja, o vértice de onde a aresta sai.

Para fins de simplificação, alguns detalhes foram omitidos do algoritmo. Por exemplo, no NMESIS a segunda parte (entre a linha 10 e 14) é executada apenas para problemas com mais de três dimensões ($D > 3$). Para problemas menores, apenas a execução do NBC já é suficiente, ou até mesmo uma amostragem uniforme. A verificação da linha 13 é feita com o *Hill-Valley*, que utiliza $nint_{ini} = \lfloor \log(D) \rfloor$, se $D \geq 3$, onde $nint_{ini}$ é o número de pontos amostrados sobre a aresta que conecta os vértices adjacentes.

Os vértices não-dominados, ou “sementes”, são utilizados para definir a posição inicial de cada nicho. Através dessa inicialização é possível obter tanto o raio do nicho como a população inicial de cada CMA-ES. Após os centroides serem definidos, deve-se agrupar indivíduos de acordo com a informação *a priori* do número de grupos e seus centroides. Quando um nicho não atinge o mínimo de indivíduos imposto pela configuração do CMA-ES, o restante é gerado utilizando o passo de mutação do CMA-ES para completar a população. Na Figura 3.2.1 o resultado utilizando somente o NBC (esq.) e depois de aplicar a extensão com *Hill-Valley* (NBC+DM) na segunda figura (dir.). Nota-se que o NBC+DM consegue separar corretamente o grafo, recuperando praticamente todas as regiões de atração (11 de 12).

3.2.2 *Niching*

A subseção anterior tratou de como o algoritmo inicializa a população e os motivos por trás dessas decisões. Nesta subseção, serão tratados os aspectos referentes à propriedade de *niching* do algoritmo.

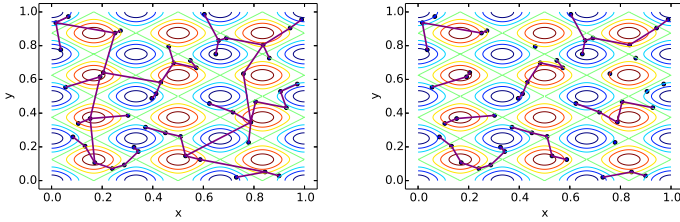


Figura 8 – À esquerda, o agrupamento feito através do NBC somente; À direita, o agrupamento utilizando o NBC+DM com 1 ponto interno.

O NMESIS tem como fundação as técnicas de *niching* paralelas e explícitas, como os trabalhos de Ursem (1999), Shir, Emmerich e Bäck (2010), Preuss (2010), Fieldsend (2014) e Dong e Zhou (2014). Alguns desses métodos também utilizam o CMA-ES e portanto o método proposto compartilha semelhanças com esses métodos.

Como o NMESIS é um método explícito, é preciso definir o que é um nicho e como ele se comporta de maneira explícita. A ideia é isolar o nicho e somente permitir troca de informação com o mundo exterior quando assim for pertinente. Isso significa que trocas que possam fragmentar o nicho ou mesmo, degenerar a matriz de covariância não são aceitas.

Cada nicho é representado pela seguinte tupla de valores

$$q_k = (\mathcal{N}_k^D, \lambda_k, \sigma_k) \mid D > 0, \lambda_k > 0, \sigma_k \geq 0 \quad (3.20)$$

onde \mathcal{N}_k^D é uma distribuição normal multivariada com D parâmetros, com média $\bar{\mu}_k$ e uma matriz de covariância Σ_k . O λ_k é o tamanho da população mantida por aquele nicho e σ_k é o passo evolutivo do CMA-ES. No caso do NMESIS, o σ_k também é utilizado para construir o valor do raio do nicho. Alguns parâmetros foram omitidos pois eles são iguais para todos os nichos, como o caso dos parâmetros padrão do CMA-ES (ver Subseção 2.2.1). A arquitetura do algoritmo é apresentada na Figura 9. Observe que cada CMA-ES indicado na figura é composto por uma tupla de valores (q_k) como indicado acima.

Após a inicialização, descrita na Subseção 3.2.1, as “sementes” são utilizadas como centroides, $\bar{\mu}_k$, e o σ_k é calculado a partir do tamanho da maior aresta que conecta o centroide a um vértice adjacente pertencente a sua população. A distância (d_{nb}) obtida é tomada como duas vezes o desvio padrão da distribuição do nicho, portanto o cálculo para obter o tamanho do passo é dado pela Equação 3.21.

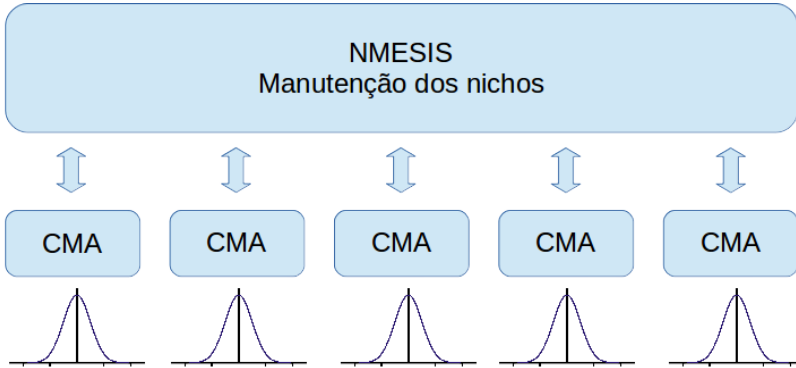


Figura 9 – Arquitetura do NMESIS. Um controle central que aplica os operadores enquanto cada nicho executa suas operações de evolução utilizando o CMA-ES.

$$\sigma_k = \frac{d_{nb}^k}{2\sqrt{n}}, \quad (3.21)$$

onde d_{nb}^k é o tamanho da maior aresta que liga μ_k a um vértice adjacente, no grafo obtido durante a inicialização pelo método NBC+DM.

Após finalizar a inicialização, o algoritmo dá início a um processo de geração-atualização. No passo de geração, cada nicho é convocado a produzir uma nova geração de soluções, utilizando o mecanismo de amostragem do CMA-ES. Cada nicho possui a sua própria instância do CMA-ES de maneira que sua população é amostrada independente dos demais nichos.

No passo de atualização, todos os indivíduos já foram avaliados pela função objetivo e possuem um valor de aptidão associado. Nessa fase, cada nicho k atualiza seus parâmetros (q_k) utilizando sua população atual. A distribuição de probabilidade da população é modificada para representar seu estado atual. O esperado é que a variância diminua com o passar do tempo, indicando a convergência do algoritmo.

Entretanto, esses dois passos são comuns a grande maioria dos algoritmos de otimização global. A diferença são dois operadores adicionados, um para coibir a execução de algum nicho e outro para gerar novos nichos.

O operador utilizado para minimizar o número de gaussianas, referenciado por **gmm_minimize**(Q, τ) no Algoritmo 5, serve como um filtro para acelerar a eliminação de nichos que estejam competindo pelo mesmo local na superfície de busca. Essa eliminação é feita através da Regra de Decisão

Bayesiana discutida na Seção 3.1. O limiar de decisão é gerado através do parâmetro τ que fornece a tolerância que será usada para calcular o θ . As Equações 3.1 e 3.2 são combinadas no operador apresentado no Algoritmo 3, linha 6 sob o nome de $aceita(q)$.

Algoritmo 3 $gmm_minimize(Q, \tau)$

Entrada Q, τ

- 1: $m := 0$
 - 2: $\theta := \mathbf{F}^{-1}(\tau)$
 - 3: ordena Q pelo valor de aptidão
 - 4: **para todo** $q_k \in ativos(Q) \wedge q_j \in Q \wedge q_k \neq q_j$ **faça**
 - 5: **se** $q_k.aceita(q_j) \wedge q_j.aceita(q_k)$ **então** ▷ Equação 3.10
 - 6: **se** q_k **não** está separado de q_j **então**
 - 7: $remove_pior(Q, q_k, q_j)$
 - 8: $m = m + 1$
 - 9: Atualiza(Q)
 - 10: **retorna** $\{Q, m\}$
-

No algoritmo são utilizadas as seguintes definições:

1. A função $ativos(Q)$ retorna todos os nichos que ainda estão ativos, i.e. não convergiram;
2. $q_j.aceita(q_k)$ representa a regra de decisão estabelecida na Equação 3.10;
3. “está separado” é a aplicação do *Hill-Valley* para detectar multimodalidade;
4. A função **remove_pior**(Q, q_k, q_j), remove do conjunto Q a pior solução entre q_k e q_j ;
5. Atualiza(Q) representa todas as alterações e checagens que são dependentes da implementação, para evitar comparar q_k com q_j duas vezes.

O operador descrito permite eliminar buscas redundantes, que tornariam o método mais lento. Além disso, o resultado pode ser prejudicado, pois normalmente métodos de otimização são executados com um limite máximo para o número de iterações e avaliações de função. Se o método gastou com buscas redundantes, a eficiência fica comprometida uma vez que a relação iterações por solução ótima aumenta. A verificação é controlada pelo parâmetro ϕ que é a quantidade de deslocamento do centroide desde o início, se $\|\mu_k^{(g)} - \mu_k^{(0)}\|^2 \leq \phi$ então o nicho k pode ser eliminado do grupo retornado

pela função *ativos*(Q) (Linha 4, Algoritmo 3). O valor padrão de ϕ utilizado nessa dissertação é $\frac{1}{3}\sigma_k^{(0)}\sqrt{D}$, que é um terço da distância referente ao tamanho do passo na geração g . Note que o nicho k fica impedido de iniciar verificações, no entanto, outros nichos podem verificá-lo, pois ele ainda pertence ao conjunto Q .

Para que um nicho seja eliminado, o Algoritmo 3 primeiramente verifica se dois nichos estão se sobrepondo (Linha 5). Caso a sobreposição seja constatada, executa-se a detecção de multimodalidade (*Hill-Valley*) para analisar a topologia entre os dois nichos. Se um “vale” for encontrado, ou seja, ambos estão convergindo para “picos” (soluções ótimas) **diferentes**, a eliminação não ocorre. Esse mecanismo permite que o algoritmo possa gerar nichos para soluções diferentes porém próximas.

No operador **gmm_minimize**(Q, τ), ao verificarmos se dois centroides estão separados ou não (Linha 5, Algoritmo 3), o algoritmo testa se o ponto médio usado pelo *Hill-Valley* é melhor que ambos os centroides em questão. Se este for caso, a função **remove_pior**(Q, q_k, q_j), não irá remover apenas o pior, mas unir os melhores de ambas as populações e usar o ponto médio como novo centroide e a combinação das duas matrizes de covariância (C_k e C_j), com peso maior para a matriz do nicho com valor de aptidão maior, de acordo com o passo evolutivo do CMA-ES (Equação 2.5) reproduzido abaixo e adaptado para os índices k e j , supondo que a aptidão de k é maior.

$$\mathbf{C} = (1 - c_1 - c_\mu)\mathbf{C}_k + c_1(\mathbf{p}_c\mathbf{p}_c^T + \delta(h_\sigma)\mathbf{C}_j) + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T \quad (3.22)$$

Um nicho pode ser eliminado por três motivos: (1) através da minimização apresentada acima; (2) convergência e (3) estagnação ou divergência. Os dois últimos casos são discutidos na Subseção 3.2.3. Embora o método de inicialização vise estimar o número de nichos, o NMESIS não pode depender apenas dessa estimativa. Para ser dinâmico com relação ao número de soluções ótimas presentes no problema, o método precisa de um mecanismo de busca global, capaz de amostrar fora dos nichos e explorar regiões ainda inexploradas, principalmente quando os nichos começam a serem eliminados.

O operador **gmm_maximize**($Q, \tau, \lambda, n_{min}$) (Algoritmo 4) é a maneira que o NMESIS utiliza para manter sempre um número ativo de nichos em execução. O parâmetro n_{min} é o número mínimo de nichos ativos que devem coexistir ao mesmo tempo por iteração. Quanto maior for esse valor, mais intensa é a busca e consequentemente, mais iterações são gastas. A cada iteração, esse operador decide se deve realizar uma amostragem geométrica ou um cruzamento binário entre dois nichos ativos. O ponto amostrado com maior aptidão, e que não foi aceito (RDB) por nenhum outro nicho, torna-se

um novo nicho. O raio deste novo nicho é calculado usando a distância de seu centroide para o nicho mais próximo e aplicando a Equação 3.21 para obter o σ_k do CMA-ES. Se o número mínimo de nichos não for atingido, o processo deve ser repetido.

As duas técnicas de amostragem utilizadas servem para explorar a estrutura do problema de maneiras diferentes. Ambas tem o intuito de encontrar regiões ainda não exploradas, no entanto cada problema possui suas peculiaridades que podem servir de armadilhas para algumas técnicas de amostragem, enquanto que para outras, esses detalhes são facilmente aproveitados. É o caso do cruzamento binário que explorar simetrias e outras relações lineares entre os nichos existentes. Enquanto que a amostragem geométrica visa explorar uniformemente o domínio do problema. Para o *benchmark* CEC 2013, foi definido que cada técnica teria 50% de chances de ser escolhida em cada iteração. No entanto, quando se trata de problemas reais e o usuário tem um conhecimento prévio do problema, ele pode privilegiar alguma das técnicas aumentando o peso da técnica que for mais apropriada. No Algoritmo 4 o peso foi fixado em 0.5.

Algoritmo 4 `gmm_maximize`($Q, \tau, \lambda, n_{min}$)

Entrada $Q, \tau, n_{min}, \lambda$

```

1:  $m := 0$ 
2:  $\theta := \mathbf{F}^{-1}(\tau)$ 
3: Enquanto  $|Q| < n_{min}$  faça
4:    $\alpha := \text{unif}(0, 1)$ 
5:   se  $\alpha > 0.5$  ou  $|Q| \leq 1$  então
6:      $W = \text{lhs}(\lambda)$ 
7:   senão
8:      $W = \text{crossover}(\lambda, Q)$ 
9:    $b := \text{seleciona}(W, Q, \theta)$ 
10:   $Q := Q \cup b$ 
11:   $m := m + |W|$ 
12: retorna  $\{Q, m\}$ 

```

No Algoritmo 4, temos os dois métodos de amostragens utilizados, a amostragem geométrica ($\text{lhs}(\lambda)$) que utiliza a amostragem por hipercubo latino (*Latin Hypercube Sampling* - LHS) (HELTON; DAVIS, 2003) para gerar pontos uniformemente distribuídos. Já a função $\text{crossover}(\lambda, Q)$ representa o cruzamento binário entre duas soluções presentes no conjunto Q . O λ é a quantidade de amostras realizadas, que é igual ao tamanho da população padrão do CMA-ES. Os símbolos presentes no algoritmo são:

1. W é a população amostrada pelo método escolhido;
2. Q é o conjunto de nichos;
3. $unif(a, b)$ é amostragem uniforme comum;
4. θ é o limiar da Regra de Decisão Bayesiana (RDB) utilizada;
5. $seleciona(W, Q, \theta)$ é a função que seleciona o melhor indivíduo que não está presente no raio de nenhum nicho;
6. m é o número de avaliações de função realizadas;

O Algoritmo 5 é uma visão simplificada do NMESIS que descreve o posicionamento dos operadores no algoritmo. O mesmo método de amostragem geométrica utilizado no Algoritmo 4 é utilizado para gerar as primeiras amostras que serão filtradas pelo Algoritmo 2 para selecionar os nichos e a população inicial de cada um deles. O tamanho dessa primeira população é representado por $\Lambda^{(0)}$. O método em seguida começa o ciclo de geração e atualização das soluções. A principal diferença desta parte, comparada a outros algoritmos de otimização canônicos, é que ele trabalha com múltiplas populações. O método executa esse ciclo até que algum critério de parada seja atingido, seja por convergência, estagnação, número máximo de avaliações da função objetivo ou número máximo de iterações. Na Subseção 3.2.3 apresentada a seguir descreve como o NMESIS define esses critérios de parada.

A função de amostragem ($Amostra(q_k)$) no Algoritmo 5 é a Equação 2.1 do CMA-ES, adaptada abaixo,

$$y_i^{(g+1)} = x^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}), \quad i = 0, 1, 2, \dots, N, \quad (3.23)$$

No algoritmo principal, a população inicial também é construída utilizando uma amostragem geométrica ($lhs(\Lambda^{(0)})$), com o método de hipercubo latino (HELTON; DAVIS, 2003). Em seguida, são extraídas as “sementes” ($\psi^{(0)}$) que são utilizadas para construir os nichos. Em $\psi^{(0)}$, está o grafo necessário para extrair os centroides e os raios dos nichos. Mais ao final do método, temos a função $marcar_convergidos(Q)$, que faz exatamente o que o nome propõe, marca os nichos que convergiram para que estes sejam evitados pela função $ativos(Q)$ utilizada na Linha 4 do Algoritmo 3. Na Figura 10, o algoritmo é executado para o problema $F11$ do *benchmark* CEC 2013. Após 44 gerações ($g = 44$) o algoritmo encontrou todas as soluções. A primeira imagem é da quarta geração, que mostra como a inicialização ajudou o método a iniciar próximo da maioria das soluções ótimas. Os pontos pretos indicam soluções, enquanto que os vermelhos representam os centroides dos nichos. Na

Algoritmo 5 NMESIS

Entrada $\Lambda^{(0)}, \tau, n_{min}, N_{max}$
Verifique $\Lambda^{(0)} > 0$

$$0,5 \leq \tau \leq 0,95$$

$$n_{min} \geq 2$$

$$N_{max} > 0$$

1: $P := lhs(\Lambda^{(0)})$

2: $\psi^{(0)} := NBC + DM(P)$

3: $Q^{(0)} :=$ Inicia Nichos com $\psi^{(0)}$

4: $m := 0$

5: $g := 0$

6: **Enquanto** $m < N_{max}$ **faça**

7: **para todo** $\{q_k \in Q^{(g)} \mid 0 \leq k \leq |Q|\}$ **faça**

8: $P_k^{(g)} := Amostra(q_k)$ ▷ Passo de geração do CMA, Equação 2.1

9: $Atualiza(q_k, P_k^{(g)})$ ▷ Atualização do CMA - ver Algoritmo 1

10: $m := m + |P^{(g)}|$

11: Verifica critérios de parada

12: $Q := marcar_convergidos(Q)$ ▷ Desativa nichos convergidos

13: $Q := Q -$ Nichos estagnados

14: $\{Q, mm\} := gmm_minimize(Q, \tau)$

15: $\{Q, mm\} := gmm_maximize(Q, \tau, n_{min})$

16: $g := g + 1$

17: $m := m + mm$

última imagem (à dir. inferior), as estrelas vermelhas indicam todos os ótimos presentes na função.

3.2.3 Validação do Nicho

As seguintes condições devem ser satisfeitas para que o nicho continue sendo atualizado. Estas condições são as mesmas utilizadas pelos algoritmos que aplicam estratégias de reinicialização, como o BIPOP-CMAES (HANSEN, 2009).

1. A razão entre o $\sigma^{(g)}$ da geração g e o $\sigma^{(0)}$ inicial é comparada com a informação contida nos autovalores,

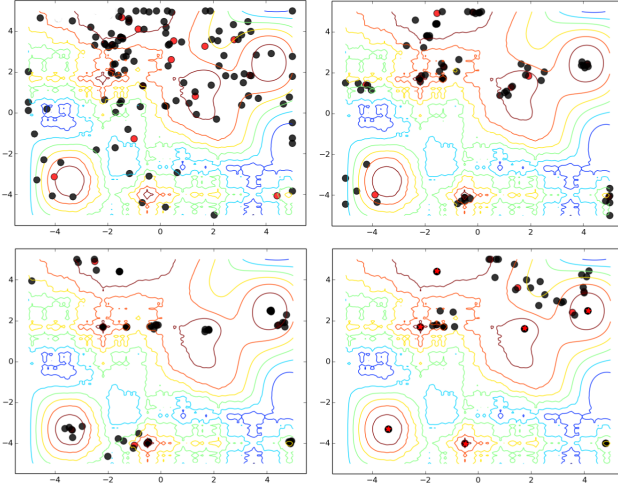


Figura 10 – Quatro gerações capturadas durante a execução do método. Na ordem a 4, 16, 24 e 40. Os pontos pretos indicam as soluções enquanto os pontos vermelhos indicam o centroide do nicho. Na última, as estrelas vermelhas indicam o local dos ótimos globais. Todos foram encontrados.

$$\begin{cases} \text{divergiu,} & \text{se } \frac{\sigma^{(g)}}{\sigma^{(0)}} > (\text{diag}D^{(g)})_n * \text{tol}_\sigma, \\ \text{continua,} & \text{senão.} \end{cases}, \quad (3.24)$$

aonde $\text{diag}D^{(g)}_n$ é o último valor do vetor $\text{diag}D$ que contém a raiz quadrada dos autovalores resultantes da decomposição da matriz de covariância $C^{(g)}$. O limiar tol_σ é definido em $1e20$.

- Quando o eixo da matriz de covariância não gera mais uma mutação considerável,

$$\begin{cases} \text{estagnado,} & \text{se } \bar{x}^{(g)} = \bar{x}^{(g)} + 0.2 * \sigma^{(g)} * \text{diag}(C^{(g)}). \\ \text{ativo,} & \text{senão.} \end{cases} \quad (3.25)$$

aonde $\text{diag}(C^{(g)})$ é a diagonal da matriz de covariância, $\sigma^{(g)}$ é o tamanho do passo e $\bar{x}^{(g)}$ é o centroide.

- O número de condição da matriz C , $\text{cond} = \frac{d_n}{d_0}$, aonde d_0 é o menor autovalor do vetor $\text{diag}D$ já ordenado. Se $\text{cond} > 10^4$ o nicho é removido, significando que o algoritmo divergiu.

4. Razão entre a variância da aptidão da população e a aptidão do centroide deve ser maior que $tol_c = 1e - 6$.
5. Quando a Equação 3.26 resultar em $\delta = 1$ o método convergiu.

$$\delta = \begin{cases} 1 & \text{Se } \frac{\text{var}(\text{fitness}(\mathbf{P}^{(g)}))}{\text{fitness}(\vec{m}^{(g)})} < 1e^{-8} \\ 0 & \text{senão} \end{cases} \quad (3.26)$$

onde $\vec{m}^{(g)}$ é o centroide e $\mathbf{P}^{(g)}$ é a matriz $\Lambda^{(g)} \times D$, que representa a população na geração g .

A última condição define quando um nicho convergiu para uma única solução, perdendo toda a diversidade. Nesta situação o nicho continua ativo mas na prática a execução de um operador jamais irá partir de um nicho que já convergiu. Isso ocorre por dois motivos: (1) O nicho perdeu toda a diversidade e o seu raio é praticamente nulo, coibindo qualquer tentativa de aplicar um verificação na sua vizinhança; (2) Após convergir ele não terá mais participação ativa em nenhuma sobreposição, pois ele não será mais atualizado. Contudo ele continuará podendo ser sobreposto e/ou avaliado por outros nichos.

3.3 DISCUSSÃO

Os algoritmos de otimização multimodal são métodos destinados a encontrar múltiplas soluções. Situações reais geralmente requerem diferentes opções para tratar um problema, sendo essa a principal motivação da otimização multimodal. No entanto estes métodos precisam ser robustos, precisos e eficientes para serem utilizados na prática.

No Seção 2.6, este trabalho abordou os algoritmos que, como o NEA2, foram especialmente desenvolvidos para explorar problemas multimodais eficientemente. Embora todos os métodos tenham o objetivo principal de aumentar a taxa de acerto e a precisão, eles ainda apresentam um desempenho aquém do esperado para problemas com mais de cinco parâmetros (FIELDSEND, 2014).

A grande dificuldade dos métodos de otimização multimodal, está no tamanho do problema (ou número de dimensões). Quanto maior o problema, menos informação útil o algoritmo consegue obter, uma vez que o tamanho do espaço de busca cresce exponencialmente com relação ao número de dimensões. Aumentar a população amostrada para compensar o tamanho do problema torna-se computacionalmente inviável.

O NMESIS, apresentado neste capítulo, é um método de otimização multimodal que alia a modelagem da população como uma mistura de gaussianas com uma inicialização da população específica para problemas multi-

modais. Esta inicialização permite ao método executar as buscas em lugares com maior probabilidade de conterem um ótimo global. Já a modelagem serve para permitir o uso de RDB e distância de Mahalanobis para verificação de similaridade, pois são métodos mais apropriados quando a população é representada por uma distribuição normal. Ambas as técnicas permitem a redução no número de avaliações de função objetivo (FO), uma vez que as buscas ficam mais concentradas e sem redundâncias. O uso do *Hill-Valley* também reduz a dependência do método com relação a métrica de distância.

O próximo capítulo trata dos experimentos realizados com o *benchmark* CEC 2013. Esse *benchmark* permite a comparação com outros métodos que também utilizaram o mesmo protocolo de experimento.

4 EXPERIMENTOS

4.1 BENCHMARK CEC 2013

O CEC 2013 ¹ é o *benchmark* mais recente desenvolvido para comparar algoritmos multimodais (LI; ENGELBRECHT; EPITROPAKIS, 2013a) (LI; ENGELBRECHT; EPITROPAKIS, 2013b). Composto por 20 problemas (12 funções mais 8 variações), abordando diferentes características para a avaliação de algoritmos de *niching*. Os primeiros 10 problemas são simples e bem conhecidos, sendo largamente utilizados em trabalhos recentes (LI, 2010; DEB; SAHA, 2010). O restante das problemas foram construídas segundo o paradigma de composição de funções (SUGANTHAN et al., 2005) e portanto são mais complexos. O *benchmark* também indica algumas métricas que podem ser utilizadas para comparar os resultados.

Os problemas presentes são compostos por 12 funções:

- F_1 : Five-Uneven-Peak Trap (1D)
- F_2 : Equal Maxima (1D)
- F_3 : Uneven Decreasing Maxima (1D)
- F_4 : Himmelblau (2D)
- F_5 : Six-Hump Camel Back (2D)
- F_6 : Shubert (2D, 3D)
- F_7 : Vincent (2D, 3D)
- F_8 : Rastrigin modificada - Todos os ótimos são globais (2D)
- F_9 : Função composta 1 (2D)
- F_{10} : Função composta 2 (2D)
- F_{11} : Função composta 3 (2D, 3D, 5D, 10D)
- F_{12} : Função composta 4 (3D, 5D, 10D, 20D)

Características:

1. Todos foram formulados como problemas de maximização.

¹Fonte:<http://goanna.cs.rmit.edu.au/xiaodong/cec13-niching/>

2. F_1 , F_2 e F_3 são simples problemas multimodais 1D.
3. F_4 e F_5 são simples problemas multimodais 1D. Estas funções não são escaláveis.
4. F_6 até F_8 são funções multimodais escaláveis. O número de ótimos da F_6 e F_7 é determinado pelo número de dimensões D . Já a função F_8 é independente do número de dimensões e pode ser controlada pelo usuário.
5. F_9 até F_{12} são funções multimodais escaláveis construídas a partir de outras funções multimodais. F_9 e F_{10} são separáveis, e assimétricas, enquanto F_{11} e F_{12} são não-separáveis e assimétricas. O número de ótimos globais é independente de D e pode ser controlado pelo usuário.

4.1.1 Métricas de Performance

Um dos critérios de comparação entre os diferentes algoritmos de *niching* é a capacidade de localizar todos os ótimos globais. Para conseguir definir esse critério deve-se primeiramente especificar a precisão necessária para considerar que a resposta foi encontrada (e.g. $0 < \varepsilon \leq 1$). As seguintes informações estão disponíveis:

- O número de ótimos globais;
- A função objetivo que define os máximos globais;
- O tamanho mínimo do raio do nicho r (menor distância entre dois ótimos); globais que estejam muito próximos.

Estas informações são necessárias para determinar o número de ótimos globais encontrados. Basicamente o *benchmark* recebe, ao final de cada iteração, um conjunto de soluções L_{sorted} já ordenado pelos valores de aptidão. Esse conjunto é então avaliado para verificar se existem ótimos globais no conjunto, e em caso de existir mais de um, também é verificado se eles são semelhantes. A verificação de similaridade é feita utilizando o parâmetro r (Tabela 2), caso duas soluções estiverem a uma distância menor que r , a solução com aptidão menor é removida. O parâmetro r é fornecido pelo *benchmark* e é uma distância absoluta.

O resultado é uma lista de soluções que estão a uma distância mínima de r umas das outras e portanto foram consideradas soluções diferentes. Definir corretamente o r é extremamente importante para evitar contagens incorretas de ótimos globais. Os valores utilizados neste *benchmark* estão definidos

na Tabela 2. Dado que se conhece os valores exatos dos ótimos globais *a priori*, é possível comparar os algoritmos em termos de *razão de picos*, *taxa de sucesso* e *número médio de avaliações da função objetivo*. Para manter uma compatibilidade com o algoritmo do *benchmark* as abreviações são referentes aos nomes em inglês. *Razão de Acertos (Razão de Acertos - RA)* é calculado utilizando a Equação 4.1:

$$RA = \frac{\sum_{run=1}^{NR} NPF_i}{NKP \times NR} \quad (4.1)$$

onde NPF_i denota o número de ótimos globais encontrados ao final da i -ésima execução, NKP é o número de ótimos globais conhecidos mostrado na Tabela 2 e NR é o número total de execuções. RA mede a porcentagem média de ótimos globais encontrados em NR execuções.

A *Taxa de Sucesso (Taxa de Sucesso - TS)* é a porcentagem de execuções bem sucedidas dentre todas as execuções (NR), dado seguinte Equação,

$$TS = \frac{NSR}{NR} \quad (4.2)$$

onde NSR é o número de execuções bem sucedidas. Um execução bem sucedida ocorre quando todos os ótimos globais são encontrados.

A *Velocidade Média de Convergência (VMC)* é obtida contando a quantidade de vezes que a função objetivo foi avaliada para localizar todos os ótimos globais, dada uma certa precisão, indicada por ϵ . É feita a média entre todas as execuções realizadas.

$$VMC = \frac{\sum_{n=0}^{NR} NFA_i}{NR} \quad (4.3)$$

aonde NFA_i indica o número de avaliações da função objetivo utilizadas na i -ésima execução. $MaxFAs$ será utilizado ao calcular a média caso não forem encontrados todos os ótimos do problema.

4.1.2 Configuração do Experimento

O NMESIS é comparado com o resultado de outros 5 métodos de otimização multimodal que foram avaliados utilizando o mesmo *benchmark* com 20 problemas da “Competição de Métodos de Niching para Otimização Multimodal” do CEC 2013 (LI; ENGELBRECHT; EPITROPAKIS, 2013a). Seguindo o mesmo protocolo utilizado na competição, os resultados obtidos

pelo NMESIS podem ser diretamente comparados aos resultados produzidos sobre as mesmas condições. O *benchmark* é independente de máquina, avaliando primeiramente a quantidade de soluções encontradas, mas também a taxa de sucesso e o número de avaliações da função objetivo. Os resultados foram comparados com o BIPOP-CMA-ES (HANSEN, 2009), NMMO (FIELDSEND, 2014), dADE (EPITROPAKIS; LI; BURKE, 2013) e o NEA2 (PREUSS, 2012) (ganhador do CEC 2013). Também é feita a comparação com os resultados do TN- CMA-ES (PEREIRA; NETO; ROISENBERG, 2014) que é a versão anterior do NMESIS, para avaliar a importância das modificações e comparar com outro método que utiliza *Hill-Valley* e o CMA-ES. Os parâmetros usados para cada algoritmo foram os mesmo utilizados em suas publicações originais. Os critérios de avaliação estão descritos na Tabela 2. O parâmetro r fornece a distância máxima (no espaço de soluções) que uma solução deve estar de um “pico” (solução ótima máxima) para que seja reconhecido que ela “encontrou” um ótimo, de acordo com um nível de precisão definido por ϵ , que é erro aceito. Para todos os problemas cinco níveis de precisão diferentes são utilizados: $\epsilon = \{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$. O código usado para o *benchmark* foi disponibilizado pelos organizadores da competição CEC 2013.

O NMESIS utiliza a tolerância de classificação $\tau = 2$, o tamanho da população do nicho $\lambda = 10D$ (aonde D é o número de dimensões), a tolerância de fusão $tol_{kl} = 1$, $nmax = inf$ e o número máximo de nichos atualizados por iteração $max_{inc} = 10$.

Os parâmetros acrescentados pelo NMESIS estão descritos na Tabela 3. Os mesmo parâmetros foram utilizados em todas as funções, com variações determinadas pelo número de parâmetros do problema (D). A configuração do CMA-ES, $param_{cma}$, foi estudada por Hansen e Ostermeier (2001) e aprimorada para possuir desempenho semelhante em diferentes problemas sem a necessidade de ajustes (HANSEN; MÜLLER; KOUMOUTSAKOS, 2003; HANSEN, 2005), portanto os valores padrão serão utilizados.

4.1.3 Resultados

Para facilitar a comparação entre os algoritmos, seus respectivos resultados foram compilados nas Figuras 11 e 12. O *benchmark* foi aplicado em cinco níveis de precisão diferentes, de $1e-1$ até $1e-5$. Como os níveis de precisão intermediários não apresentaram nenhuma característica que valesse ser discutida, para fins de simplificação, apenas os resultados do primeiro e último nível serão discutidos. Essa diferença já é o suficiente para discutirmos as características presentes em cada método. Entretanto, na Tabela 4, o

Tabela 2 – Parâmetros das funções usados para medir a performance.

Índice	Função	r	F*	Nº de ótimos globais (NKP)
1	$F1(1D)$	0.01	200.0	2
2	$F2(1D)$	0.01	1.0	5
3	$F3(1D)$	0.01	1.0	1
4	$F4(2D)$	0.01	200.0	4
5	$F5(2D)$	0.5	1.03163	2
6	$F6(2D)$	0.5	186.7309	18
7	$F7(2D)$	0.2	1.0	36
8	$F6(3D)$	0.5	2709.0935	81
9	$F7(3D)$	0.2	1.0	216
10	$F8(2D)$	0.01	-2.0	12
11	$F9(2D)$	0.01	0.0	6
12	$F10(2D)$	0.01	0.0	8
13	$F11(2D)$	0.01	0.0	6
14	$F11(3D)$	0.01	0.0	6
15	$F12(3D)$	0.01	0.0	8
16	$F11(5D)$	0.01	0.0	6
17	$F12(5D)$	0.01	0.0	8
18	$F11(10D)$	0.01	0.0	6
19	$F12(10D)$	0.01	0.0	8
20	$F12(20D)$	0.01	0.0	8

resultado completo obtido pelo NMESIS é apresentado.

Através dos gráficos apresentados nas Figuras 11 e 12, é visível a consistência dos resultados apresentados pelo NMESIS em comparação com os outros métodos. Na Figura 11, no gráfico referente a precisão de $1e^{-1}$, observa-se uma certa equivalência entre o NEA2, NMMSO, NMESIS e dADE. Nesse nível de precisão, é esperado que os métodos no estado da arte desempenhem bem, principalmente aqueles com um mecanismo de busca global mais avançado como o dADE. Contudo, na precisão de $1e^{-5}$, os métodos providos de um mecanismo de busca local tendem a produzir resultados melhores. O gráfico da Figura 11.b confirma essa afirmação, mostrando que os métodos que conseguiram manter resultados consistentes foram o NEA2, NMMSO e NMESIS. O último demonstrando variar muito pouco com relação ao gráfico da Figura 11.b. Esses resultados demonstram que, com relação aos problemas do *benchmark*, o NMESIS possui resultados consistente, indicando que não é dependente da aleatoriedade do processo de amostragem para encontrar as soluções.

Tabela 3 – Configuração do NMESIS

Parâmetro	Descrição	Valor
λ	Tamanho da população de cada nicho	$2 \times D$
tol_c	Tolerância mínima de convergência	$1e-06$
τ	Probabilidade de pertencer ao nichos ($F_{\chi^2}^{-1}(\tau)$)	$\tau = 0.95$
$nint_{init}$	Qtde. de valores intermediários avaliados	$\lfloor \log(D) \rfloor$
$\Lambda^{(0)}$	Tamanho da população usada na inicialização	$20 \times D$

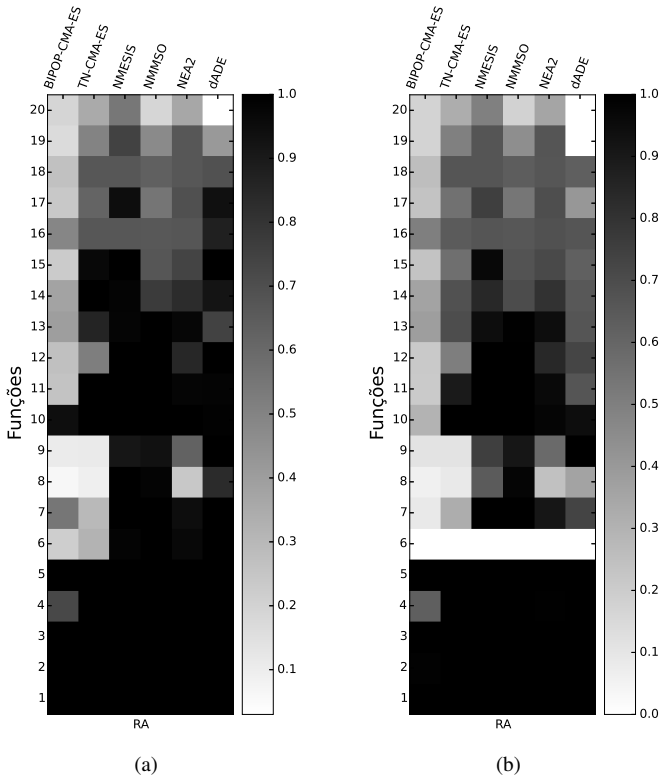


Figura 11 – Razão de Acerto (RA) para o *benchmark* CEC 2013. (a) RA de cada algoritmo para precisão de $1e-01$. (b) Idem para a precisão de $1e-05$.

O NMESIS conseguiu encontrar mais soluções que os outros concorrentes, i.e. obteve um *RA* mais alto. E principalmente, a melhoria ocorreu em todos os problemas com mais de cinco dimensões, alcançando um dos

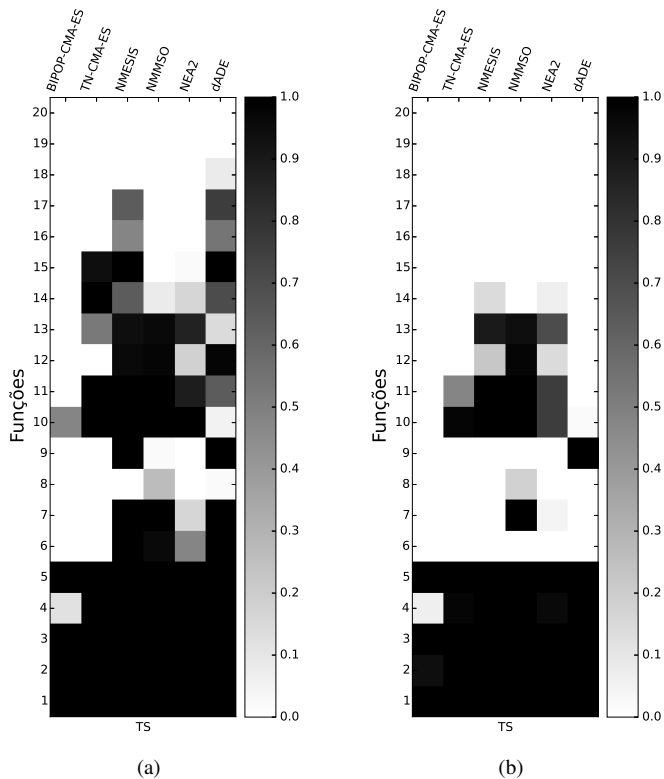


Figura 12 – Taxa de Sucesso (TS) para o *benchmark* CEC 2013. (a) TS de cada algoritmo para precisão de $1e-01$. (b) Idem para a precisão de $1e-05$.

objetivos deste trabalho. Os ganhos em termos de *RA* foram expressivos, demonstrando que o método obtém resultados consistentes e muito próximos do resultado esperado. Essa capacidade está associada a escolha do CMA-ES como método base. O CMA-ES é capaz de explorar uma região de atração com populações pequenas e de maneira bastante eficiente. Isso permite obter soluções rapidamente, economizar avaliações e como resultado, obter mais soluções pois sobram avaliações para explorá-las mesmo quando a precisão requerida é alta.

Na Tabela 5 e na Tabela 6 estão apresentadas as a taxa de convergência do NMESIS, NMMSO, dADE, TN-CMA-ES e BIPOP-CMA-ES para a precisão de $1e-01$ e $1e-04$ (as taxas obtidas pelo NEA2 na competição não foram publicadas). A escolha dos níveis se deve, principalmente, para facilitar a

Tabela 4 – Razão de Acertos (RA) e Taxa de Sucesso (TS) do NMESIS.

Precisão ε	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	RA	TS	RA	TS	RA	TS	RA	TS	RA	TS
$1.0e^{-1}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$1.0e^{-2}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$1.0e^{-3}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$1.0e^{-4}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$1.0e^{-5}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Precisão ε	$F_6(2D)$		$F_7(2D)$		$F_8(3D)$		$F_7(3D)$		$F_8(2D)$	
	RA	TS	RA	TS	RA	TS	RA	TS	RA	TS
$1.0e^{-1}$	0.983	1.000	1.000	1.000	0.993	0.000	1.000	1.000	1.000	1.000
$1.0e^{-2}$	0.880	0.040	0.830	0.000	0.193	0.000	0.411	0.000	1.000	1.000
$1.0e^{-3}$	0.874	0.060	0.816	0.000	0.194	0.000	0.409	0.000	1.000	1.000
$1.0e^{-4}$	0.871	0.100	0.821	0.000	0.183	0.000	0.408	0.000	1.000	1.000
$1.0e^{-5}$	0.852	0.040	0.813	0.000	0.195	0.000	0.375	0.000	1.000	1.000
Precisão ε	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	RA	TS	RA	TS	RA	TS	RA	TS	RA	TS
$1.0e^{-1}$	1.000	1.000	1.000	1.000	0.963	1.000	1.000	0.760	0.963	1.000
$1.0e^{-2}$	1.000	1.000	1.000	0.960	0.963	1.000	1.000	0.640	0.963	0.760
$1.0e^{-3}$	1.000	1.000	1.000	0.360	0.963	1.000	1.000	0.420	0.963	0.460
$1.0e^{-4}$	1.000	0.960	1.000	0.320	0.963	1.000	1.000	0.360	0.963	0.000
$1.0e^{-5}$	1.000	0.940	1.000	0.320	0.963	1.000	1.000	0.240	0.963	0.000
Precisão ε	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	RA	TS	RA	TS	RA	TS	RA	TS	RA	TS
$1.0e^{-1}$	0.673	0.660	0.944	0.760	0.667	0.000	0.667	0.000	0.492	0.000
$1.0e^{-2}$	0.670	0.640	0.893	0.660	0.667	0.000	0.667	0.000	0.492	0.000
$1.0e^{-3}$	0.670	0.000	0.792	0.000	0.667	0.000	0.667	0.000	0.497	0.000
$1.0e^{-4}$	0.670	0.000	0.783	0.000	0.667	0.000	0.667	0.000	0.495	0.000
$1.0e^{-5}$	0.667	0.000	0.782	0.000	0.667	0.000	0.667	0.000	0.495	0.000

comparação e verificar como o método se comporta com a variação da precisão requerida. Especificamente, para os problemas do *benchmark*, não ocorrem variações significativas da taxa de convergência na última precisão ($1e - 05$). Isso porque todos os métodos atingem o limite de avaliações da função objetivo (FAs) disponíveis para quase todos os problemas que são significativos em termos de comparação (aqueles acima de 3 dimensões). Além disso, os métodos NMMSO e dADE apenas disponibilizaram esses valores. Entretanto é possível validar algumas hipóteses usando apenas esses valores. Observe que tanto o dADE como o NMMSO possuem um desempenho melhor quando a precisão exigida é menor, enquanto o NMESIS desempenha melhor quando se requer mais precisão. Esse comportamento é causado justamente pelo método base dos algoritmos, que no caso, são o PSO (KENNEDY, 2010) e o DE (STORN; PRICE, 1997) para o NMMSO e dADE, respectivamente. Ambos os métodos são fortemente baseados em população, e se espalham rapidamente pela superfície de busca, entretanto sua mecânica de convergência não é tão sofisticada quanto o CMA-ES (HANSEN; MÜLLER; KOUMOUTSAKOS, 2003) utilizado pelo NMESIS.

Tabela 5 – Taxa de Convergência para os métodos NMESIS, NMMSO, dADE/nrand/1, TN-CMA-ES e BIPOP-CMA-ES. Com precisão especificada de $1e-01$. Os melhores resultados estão destacados.

1.00E - 01											
Algoritmo	Função	1	2	3	4	5	6	7	8	9	10
dADE/nrand/1	Média	5922	221	203	3107	367	27459	2911	367282	396811	3392
	Desv. Padrão	1673	38	14	845	121	6904	618	42450	22547	653
NMMSO	Média	578	167	46	1191	114	68441	27349	391589	399982	1422
	Desv. Padrão	135	62	46	947	53	42978	14198	21682	125	444
BIPOP	Média	50	844	98	47709	1000	200000	200000	400000	400000	169064
	Desv. Padrão	68	569	141	8033	863	0	0	0	0	45652
TN-CMA-ES	Média	1713	1783	2135	3490	1221	200000	200000	400000	400000	5947
	Desv. Padrão	252	192	2585	1596	268	0	0	0	0	388
NMESIS	Média	163	421	234	1374	189	72843	24457	400000	400000	1348
	Desv. Padrão	72	93	132	316	26	8745	3763	0	0	267
Algoritmo	Função	11	12	13	14	15	16	17	18	19	20
dADE/nrand/1	Média	45456	114735	182185	219869	61965	292773	200503	392376	340214	400000
	Desv. Padrão	58240	21749	47527	154230	16890	133136	127782	33324	95904	0
NMMSO	Média	5836	49537	53038	391400	400000	400000	400000	400000	400000	400000
	Desv. Padrão	2872	41905	40697	36202	0	0	0	0	0	0
BIPOP	Média	200000	200000	200000	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	0	0	0	0	0	0	0	0	0	0
TN-CMA-ES	Média	152740	64315	191875	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	1265	13589	20536	0	0	0	0	0	0	0
NMESIS	Média	4726	42672	40763	199856	123846	274897	193766	388270	270536	400000
	Desv. Padrão	854	5672	8154	87480	23784	87233	43784	14708	44077	0

4.2 DISCUSSÃO

A dificuldade de resolver problemas com muitos parâmetros ainda permanece. Nenhum dos métodos foi capaz de encontrar todas as soluções

Tabela 6 – Taxa de Convergência para os métodos NMESIS, NMMSO, dADE/nrand/1, TN-CMA-ES e BIPOP-CMA-ES. Com precisão especificada de $1e-04$. Os melhores resultados estão destacados.

1.00E-04											
Algoritmo	Função	1	2	3	4	5	6	7	8	9	10
dADE/nrand/1	Média	20202	1801	1290	12703	3567	150328	200000	393667	400000	12904
	Desv. Padrão	2788	586	565	1668	652	352209	0	17665	0	2169
NMMSO	Média	1089	487	342	1910	617	88759	31350	392525	400000	2686
	Desv. Padrão	179	124	131	913	149	40262	13175	21587	0	388
BIPOP	Média	41	21533	4906	47389	2424	200000	200000	400000	400000	169064
	Desv. Padrão	63	12885	6050	10411	1969	0	0	0	0	45652
TN-CMA-ES	Média	1775	6542	3640	18301	6776	200000	200000	400000	400000	6776
	Desv. Padrão	266	2457	2371	6589	4184	0	0	0	0	4184
NMESIS	Média	9043	512	569	1285	1670	92000	200000	400000	400000	1426
	Desv. Padrão	156	73	147	72	84	7453	3763	0	0	187
Algoritmo	Função	11	12	13	14	15	16	17	18	19	20
dADE/nrand/1	Média	200000	200000	200000	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	0	0	0	0	0	0	0	0	0	0
NMMSO	Média	9057	68049	74120	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	3261	38807	43854	0	0	0	0	0	0	0
BIPOP	Média	200000	200000	200000	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	0	0	0	0	0	0	0	0	0	0
TN-CMA-ES	Média	152740	64315	191875	400000	400000	400000	400000	400000	400000	400000
	Desv. Padrão	1265	13589	20536	0	0	0	0	0	0	0
NMESIS	Média	68743	42672	145763	199856	400000	400000	400000	400000	400000	400000
	Desv. Padrão	854	5672	43154	87480	0	0	0	0	0	0

para os problemas com mais de 5 parâmetros com erro menor que $1e-01$. Entretanto, os problemas propostos neste *benchmark*, que são a composição de várias funções, são realmente complexos de serem resolvidos, oferecendo um verdadeiro desafio aos métodos avaliados. Embora as modificações propostas neste trabalho não tenham produzido efeito significativo na taxa de sucesso e na velocidade de convergência, o método obteve resultados expressivos no número de ótimos encontrados, conseguindo superar o último vencedor do CEC13 no número de ótimos encontrados no problema mais difícil, $F12(20D)$, com desempenho 1.42 vezes melhor que o NEA2. O NMESIS demonstrou maior estabilidade, mantendo os resultados mesmo com o aumento do nível de precisão ou o número de dimensões. É interessante observar que o NMESIS, que herda elementos tanto do NEA2 como do NMMSO, manteve o melhor comportamento de ambos os métodos, obtendo bom desempenho nas funções $F6(2D)$, $F7(2D)$, $F6(3D)$ e $F7(3D)$ assim como o NMMSO e com resultados melhores nas funções com $D > 5$ como o NEA2.

5 CONCLUSÃO

Nesta dissertação foi introduzido uma nova técnica de otimização multimodal que utiliza o algoritmo CMA-ES como base para o seu mecanismo de melhoramento da população de soluções. Essa técnica recebeu o nome de NMESIS, referente a designação em inglês *Niching Multipopulation Evolution Strategy with Improved Search*, ou Estratégia Evolucionária baseada em Multipopulação com busca aprimorada. O aprimoramento da busca vêm do uso de uma inicialização preparada para problemas multimodais, que foi também desenvolvida durante esta pesquisa e publicada no CEC 2014 (2014 IEEE Congress on Evolutionary Computation) (PEREIRA; NETO; ROISENBERG, 2014), que utiliza informação topológica juntamente com a distância para encontrar regiões de atração com maior probabilidade de abrigarem uma solução ótima. Além do desenvolvimento da técnica, esta dissertação se propôs a revisar as melhores técnicas existentes utilizando os resultados do *benchmark* CEC 2013 como parâmetro.

Geralmente os métodos de otimização global tem como objetivo encontrar a melhor solução dado uma função objetivo multidimensional (HOLLAND, 1975)(KIRKPATRICK, 1984). Entretanto, para determinados problemas reais apenas uma solução não é o suficiente, alguns deles são: *design* de turbinas e estruturas de aviões; *design* de estruturas usadas na construção civil; logística — alocação de recursos, agendamento e combinação (NOREIKIS; BUTKUS; NURMINEN, 2014); inversão de formas de onda sísmica (KOPER; WYSESSION; WIENS, 1999); gerenciamento de combustível (TURINSKY, 2010) e otimização do *design* de reatores nucleares (PEREIRA; SCHIRRU; MARTINEZ, 1999). Esses problemas normalmente estão relacionados a uma ideia de estrutura que um pequeno detalhe alterado na solução, continua sendo uma solução ótima porém com alguma vantagem como custo ou usabilidade. No entanto, esses problemas envolvem muitos parâmetros e os métodos atuais de otimização multimodal ainda tem dificuldade para resolver problemas desse tipo.

No Seção 2.6, este trabalho abordou os algoritmos que, como o NEA2, foram especialmente desenvolvidos para explorar problemas multimodais eficientemente. Embora todos os métodos tenham o objetivo principal de aumentar a taxa de acerto e a precisão, eles ainda apresentam um desempenho aquém do esperado para problemas com mais de cinco parâmetros (FIELDSEND, 2014).

Tendo esse tipo de problema como principal motivação, buscou-se na literatura os diversos métodos de *niching*, a evolução deles para as técnicas atuais e como as técnicas canônicas são utilizadas como base. Esse estudo, apresentado no Capítulo 2, permitiu identificar alguns pontos de melhoria.

A inicialização, a representação das populações e o isolamento das soluções foram os pontos escolhidos para serem pesquisados.

Esses três fatores culminaram no desenvolvimento do NMESIS. O método de inicialização utilizado, NBC+DM, foi desenvolvido para funcionar independentemente do número de dimensões do problema. Pois ele combina um algoritmo de agrupamento baseado em grafos com uma verificação topológica entre os os vértices adjacentes. Essa inicialização assume que regiões de atração promissoras estão separadas por, pelo menos, a média das distâncias entre todos os pontos amostrados ou por algum “vale”. O conceito de “vale” refere-se a uma solução inferior entre duas soluções boas em problemas de maximização. O isolamento foi feito de maneira que os nichos se comuniquem apenas através de operadores. Um dos operadores serve para maximizar a aptidão das soluções de maneira global, explorando o espaço de busca com novos nichos. Já o outro operador, minimiza o número de nichos, com o objetivo de reduzir o custo do algoritmo e manter apenas as melhores e mais promissoras soluções. A representação das populações foi um dos pontos mais explorados no algoritmo. Para desenvolver bons operadores é importante que a população seja representada de maneira que os operadores possam acessar o máximo de informação da maneira mais simples possível. Ao utilizar o CMA-ES, foi possível representar as populações dos nichos como um modelo de mistura de gaussianas (GMM). Essa representação abre um leque de alternativas que podem ser utilizadas como operadores, desde as técnicas utilizadas nesse trabalho como o BDR e a distância de Mahalanobis, como também o algoritmo de Expectativa-Maximização (*Expectation Maximization* - EM) (MOON, 1996).

O algoritmo foi avaliado usando o protocolo do CEC 2013 para problemas multimodais e comparado com outros métodos. Os resultados apontaram que o método proposto foi capaz de resolver a maioria dos problemas igualmente aos melhores métodos comparados, NEA2 e NMMSO. Os resultados indicaram melhorias na obtenção de soluções (Razão de Acerto - Figura 11) em todos os problemas, em especial naqueles com mais de cinco dimensões aonde os resultados podem ser visto mais claramente. O método também apresentou um desempenho constante com relação a precisão requerida (Tabela 4). O método ainda gasta mais avaliações de funções que o NMMSO, que demonstrou ser um método bastante econômico. No entanto, em muitas funções a diferença foi quase insignificante estatisticamente (Tabelas 5 e 6).

Construir algoritmos híbridos requer uma avaliação das características que funcionam bem nos métodos escolhidos. No caso do NMESIS as características e a mecânica dos operadores foram escolhidas de acordo com as potencialidades do CMA-ES, como por exemplo a informação oferecida pela matriz de covariância para extrair a área de cobertura do nicho. Desta

maneira foi possível obter um método eficiente, devido ao reaproveitamento dos cálculos, ao mesmo tempo que permitia o uso do CMA-ES que é um dos melhores otimizadores atuais (HANSEN, 2005, 2009; LOSHCHILOV, 2013).

Infelizmente o NMESIS ainda não consegue obter uma escalabilidade suficiente para problemas com elevado número de dimensões. Apesar de testes não terem sido feitos, é possível observar que o método não teve uma boa performance em termos de tempo de execução para os problemas com número de parâmetros maior que 10, onde o número de avaliações da função objetivo atingiu o limite sem alcançar todas as soluções. Essa era uma situação já esperada, dado que o CMA-ES tem problemas de escalabilidade devido ao custo da matriz de covariância. Entretanto, trabalhos recentes vem apresentando resultados com o CMA-ES e a informação da matriz de covariância compactada (LOSHCHILOV, 2014). Os benefícios desses estudos irão impactar diretamente na escalabilidade do NMESIS. Além disso, é possível utilizar o CMA-ES apenas com a diagonal principal, ao custo de reduzir a performance do método em alguns problemas.

Como trabalhos futuros, existe a possibilidade de continuar estudando formas de compactar a informação da matriz de covariância, utilizando operações mais simples e reduzindo a complexidade do método. Outra possibilidade é aplicar os operadores desenvolvidos neste trabalho em outros métodos de otimização ou em conjunto com modelos que substituem a função objetivo, aproveitando as avaliações de função extras que o método executa. A aplicação do NMESIS em problemas reais também é uma possibilidade, dado que existem uma gama de problemas que envolvem de 20 à 100 dimensões e que podem ser exploradas pelo NMESIS. Como o *benchmark* CEC 2013 testa diversos ambientes e complexidades, é esperado que o método consiga resolver problemas reais com a mesma capacidade que resolveu os problemas do *benchmark*. É possível também, testar o NMESIS em problemas de otimização dinâmicos, apenas removendo a restrição de que nichos convergidos ficam inativos e definindo algumas restrições para o tamanho do passo mínimo.

REFERÊNCIAS

- ALBA, E.; DORRONSORO, B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. **Evolutionary Computation, IEEE Transactions on**, IEEE, v. 9, n. 2, p. 126–142, 2005.
- ANDO, S.; SAKUMA, J.; KOBAYASHI, S. Adaptive isolation model using data clustering for multimodal function optimization. In: **Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2005. (GECCO '05), p. 1417–1424. ISBN 1-59593-010-8. Disponível em: <<http://doi.acm.org/10.1145/1068009.1068235>>.
- BAGHEL, M.; AGRAWAL, S.; SILAKARI, S. Survey of metaheuristic algorithms for combinatorial optimization. **International Journal of Computer Applications**, Citeseer, v. 58, n. 19, p. 21–31, 2012.
- BEASLEY, D.; BULL, D. R.; MARTIN, R. R. A sequential niche technique for multimodal function optimization. **Evolutionary computation**, MIT Press, v. 1, n. 2, p. 101–125, 1993.
- BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies – a comprehensive introduction. **Natural Computing**, Kluwer Academic Publishers, v. 1, n. 1, p. 3–52, 2002. ISSN 1567-7818.
- CAVICCHIO, D. J. Adaptive search using simulated evolution. 1970.
- DAS, S. et al. Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art. **Swarm and Evolutionary Computation**, v. 1, n. 2, p. 71–88, 2011.
- DEB, K.; GOLDBERG, D. E. An investigation of niche and species formation in genetic function optimization. In: **Proceedings of the 3rd International Conference on Genetic Algorithms**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989. p. 42–50. ISBN 1-55860-066-3. Disponível em: <<http://dl.acm.org/citation.cfm?id=645512.657099>>.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **Trans. Evol. Comp**, IEEE Press, Piscataway, NJ, USA, v. 6, n. 2, p. 182–197, abr. 2002. ISSN 1089-778X.
- DEB, K.; SAHA, A. Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach. In: ACM.

Proceedings of the 12th annual conference on genetic and evolutionary computation. [S.l.], 2010. p. 447–454.

DONG, W.; ZHOU, M. Gaussian classifier-based evolutionary strategy for multimodal optimization. **IEEE transactions on neural networks and learning systems**, Institute of Electrical and Electronics Engineers, v. 25, n. 6, p. 1200–1216, 2014.

EPITROPAKIS, M. G.; LI, X.; BURKE, E. K. A dynamic archive niching differential evolution algorithm for multimodal optimization. **2013 IEEE Congress on Evolutionary Computation**, Ieee, p. 79–86, jun. 2013.

FIELDSEND, J. Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser. In: **Evolutionary Computation (CEC), 2014 IEEE Congress on.** [S.l.: s.n.], 2014. p. 2593–2600.

FISHER, R. **The genetical theory of natural selection.** The Clarendon Press, 1930. Disponível em: <<http://books.google.com.br/books?id=t3jwAAAAMAAJ>>.

FLETCHER, R. **Practical methods of optimization.** [S.l.]: John Wiley & Sons, 2013.

GLOVER, F. Tabu search-part i. **ORSA Journal on computing**, INFORMS, v. 1, n. 3, p. 190–206, 1989.

GOLDBERG, D. **Genetic Algorithms in Search, Optimization, and Machine Learning.** [S.l.]: Addison-Wesley, 1989. (Artificial Intelligence). ISBN 9780201157673.

GOLDBERG, D. E.; RICHARDSON, J. Genetic algorithms with sharing for multimodal function optimization. In: **Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application.** Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987. p. 41–49. ISBN 0-8058-0158-8.

HANSEN, N. **The CMA evolution strategy: A tutorial.** [S.l.], 2005. Disponível em: <www.lri.fr/hansen/cmatutorial.pdf>.

HANSEN, N. Benchmarking a bi-population cma-es on the bbob-2009 function testbed. In: **Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers.** New York, NY, USA: ACM, 2009. (GECCO '09), p. 2389–2396. ISBN 978-1-60558-505-5.

HANSEN, N. et al. Real-parameter black-box optimization benchmarking 2010: Experimental setup. 2010.

HANSEN, N.; KERN, S. Evaluating the cma evolution strategy on multimodal test functions. In: YAO, X. et al. (Ed.). **Parallel Problem Solving from Nature - PPSN VIII**. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3242). p. 282–291. ISBN 978-3-540-23092-2.

HANSEN, N.; MÜLLER, S. D.; KOUMOUTSAKOS, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 11, n. 1, p. 1–18, mar. 2003. ISSN 1063-6560.

HANSEN, N.; OSTERMEIER, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: IEEE. **Evolutionary Computation, 1996., Proceedings of IEEE International Conference on**. [S.l.], 1996. p. 312–317.

HANSEN, N.; OSTERMEIER, a. Completely derandomized self-adaptation in evolution strategies. **Evolutionary computation**, v. 9, n. 2, p. 159–95, jan. 2001. ISSN 1063-6560.

HANSEN, N. et al. Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. **Applied Soft Computing**, Elsevier, v. 11, n. 8, p. 5755–5769, 2011.

HELTON, J. C.; DAVIS, F. J. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. **Reliability Engineering & System Safety**, Elsevier, v. 81, n. 1, p. 23–69, 2003.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1975. ISBN 0262082136.

JONG, K. A. D. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems**. Tese (Doutorado) — University of Michigan, Ann Arbor, MI, USA, 1975. AAI7609381.

KENNEDY, J. Particle swarm optimization. In: **Encyclopedia of Machine Learning**. [S.l.]: Springer, 2010. p. 760–766.

KIRKPATRICK, S. Optimization by simulated annealing: Quantitative studies. **Journal of statistical physics**, Springer, v. 34, n. 5-6, p. 975–986, 1984.

KOPER, K. D.; WYSESSION, M. E.; WIENS, D. A. Multimodal function optimization with a niching genetic algorithm: A seismological example. **Bulletin of the Seismological Society of America**, Seismological Society of America, v. 89, n. 4, p. 978–988, 1999.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **The Annals of Mathematical Statistics**, JSTOR, p. 79–86, 1951.

LI, X. Niching without niching parameters: particle swarm optimization using a ring topology. **Evolutionary Computation, IEEE Transactions on**, IEEE, v. 14, n. 1, p. 150–169, 2010.

LI, X.; ENGELBRECHT, A.; EPITROPAKIS, M. Benchmark functions for CEC 2013 special session and competition on niching methods for multimodal function optimization. **Royal Melbourne Institute of ...**, p. 1–10, 2013.

LI, X.; ENGELBRECHT, A.; EPITROPAKIS, M. Results of the 2013 ieeec cec competition on niching methods for multimodal optimization. In: **IEEE congress on evolutionary computation competition on: Niching methods for multimodal, optimization**. [S.l.: s.n.], 2013.

LIANG, J. et al. Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. **Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore**, 2014.

LIU, C. The bayes decision rule induced similarity measures. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 6, p. 1086–1090, 2007.

LOSHCHILOV, I. Cma-es with restarts for solving cec 2013 benchmark problems. In: **Evolutionary Computation (CEC), 2013 IEEE Congress on**. [S.l.: s.n.], 2013. p. 369–376.

LOSHCHILOV, I. A computationally efficient limited memory cma-es for large scale optimization. **arXiv preprint arXiv:1404.5520**, 2014.

LOSHCHILOV, I. et al. Maximum likelihood-based online adaptation of hyper-parameters in cma-es. In: **Parallel Problem Solving from Nature–PPSN XIII**. [S.l.]: Springer, 2014. p. 70–79.

MACARTHUR, R. H. **The theory of island biogeography**. [S.l.]: Princeton University Press, 1967.

- MAHFOUD, S. Genetic drift in sharing methods. In: **Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on**. [S.l.: s.n.], 1994. p. 67–72 vol.1.
- MAHFOUD, S. Niching methods for genetic algorithms. **Urbana**, n. 95001, 1995.
- MAHFOUD, S. W. A comparison of parallel and sequential niching methods. In: CITESEER. **Conference on genetic algorithms**. [S.l.], 1995. v. 136, p. 143.
- MARTIN, W. N.; LIENIG, J.; COHOON, J. P. C6.3 Island (migration) models: evolutionary algorithms based on punctuated equilibria. In: **International Conference on Evolutionary Computation**. [S.l.: s.n.], 2000.
- MOON, T. K. The expectation-maximization algorithm. **Signal processing magazine, IEEE, IEEE**, v. 13, n. 6, p. 47–60, 1996.
- MORÉ, J. J. The levenberg-marquardt algorithm: implementation and theory. In: **Numerical analysis**. [S.l.]: Springer, 1978. p. 105–116.
- MOSCATO, P. et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. **Caltech concurrent computation program, C3P Report**, v. 826, p. 1989, 1989.
- MÜHLENBEIN, H.; MAHNIG, T. Evolutionary algorithms: From recombination to search distributions. In: KALLEL, L.; NAUDTS, B.; ROGERS, A. (Ed.). **Theoretical Aspects of Evolutionary Computing**. [S.l.]: Springer Berlin Heidelberg, 2001, (Natural Computing Series). p. 135–173. ISBN 978-3-642-08676-2.
- NOREIKIS, M.; BUTKUS, P.; NURMINEN, J. In-vehicle application for multimodal route planning and analysis. In: **Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on**. [S.l.: s.n.], 2014. p. 350–355.
- ORR, H. A. Adaptation and the cost of complexity. **Evolution**, Wiley Online Library, v. 54, n. 1, p. 13–20, 2000.
- PEREIRA, C. M. do N. A.; SCHIRRU, R.; MARTINEZ, A. S. Basic investigations related to genetic algorithms in core designs. **Annals of Nuclear Energy**, Elsevier, v. 26, n. 3, p. 173–193, 1999.

PEREIRA, M. W.; NETO, G. S.; ROISENBERG, M. A topological niching covariance matrix adaptation for multimodal optimization. In: **Evolutionary Computation (CEC), 2014 IEEE Congress on**. [S.l.: s.n.], 2014. p. 2562–2569.

PETROWSKI, A. A clearing procedure as a niching method for genetic algorithms. In: **Evolutionary Computation, 1996., Proceedings of IEEE International Conference on**. [S.l.: s.n.], 1996. p. 798–803.

PETROWSKI, A.; GENET, M. G. A classification tree for speciation. In: **IEEE. Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on**. [S.l.], 1999. v. 1.

POTTER, M. A.; JONG, K. A. D. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. **Evolutionary Computation**, v. 8, p. 1–29, 2000.

PREUSS, M. Niching the CMA-ES via nearest-better clustering. **Proceedings of the 12th annual conference comp on Genetic and evolutionary computation - GECCO '10**, ACM Press, New York, New York, USA, p. 1711, 2010.

PREUSS, M. Improved topological niching for real-valued global optimization. In: **Proceedings of the 2012T European Conference on Applications of Evolutionary Computation**. Berlin, Heidelberg: Springer-Verlag, 2012. (EvoApplications'12), p. 386–395. ISBN 978-3-642-29177-7.

PREUSS, M.; SCHÖNEMANN, L.; EMMERICH, M. Counteracting genetic drift and disruptive recombination in (μ, λ) -ea on multimodal fitness landscapes. In: **In Proc. Genetic and Evolutionary Computation Conf. (GECCO 2005), Washington D.C.** [S.l.]: ACM Press, 2005. p. 865–872.

PREUSS, M.; STOEAN, C.; STOEAN, R. Niching foundations: Basin identification on fixed-property generated landscapes. In: **Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2011. (GECCO '11), p. 837–844. ISBN 978-1-4503-0557-0.

RECHENBERG, I. Evolution strategy: Nature's way of optimization. In: **Optimization: Methods and applications, possibilities and limitations**. [S.l.]: Springer, 1989. p. 106–126.

REYNOLDS, D. Gaussian mixture models. In: **Encyclopedia of Biometrics**. [S.l.]: Springer, 2009. p. 659–663.

RUNNALLS, A. R. Kullback-leibler approach to gaussian mixture reduction. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE-Institute of Electrical Electronics Engineers Inc, v. 43, n. 3, p. 989–999, 2007.

SACCO, W. F.; HENDERSON, N.; RIOS-COELHO, A. C. Topographical clearing differential evolution: A new method to solve multimodal optimization problems. **Progress in Nuclear Energy**, v. 71, n. 0, p. 269 – 278, 2014. ISSN 0149-1970. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0149197013002515>>.

SAMBRIDGE, M. A parallel tempering algorithm for probabilistic sampling and multimodal optimization. **Geophysical Journal International**, Oxford University Press, p. ggt342, 2013.

SHIR, O. M.; BÄCK, T. Niche radius adaptation in the cma-es niching algorithm. In: **Proceedings of the 9th International Conference on Parallel Problem Solving from Nature**. Berlin, Heidelberg: Springer-Verlag, 2006. (PPSN'06), p. 142–151. ISBN 3-540-38990-3, 978-3-540-38990-3.

SHIR, O. M.; EMMERICH, M.; BÄCK, T. Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. **Evolutionary computation**, v. 18, n. 1, p. 97–126, jan. 2010. ISSN 1530-9304.

STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, v. 11, n. 4, p. 341–359, 1997.

STREICHERT, F. et al. A clustering based niching ea for multimodal search spaces. In: LIARDET, P. et al. (Ed.). **Artificial Evolution**. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 2936). p. 293–304. ISBN 978-3-540-21523-3.

SUGANTHAN, P. N. et al. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. **KanGAL Report**, v. 2005005, 2005.

TURINSKY, P. J. Core isotopic depletion and fuel management. In: **Handbook of Nuclear Engineering**. [S.l.]: Springer, 2010. p. 1241–1312.

URSEM, R. Multinational evolutionary algorithms. In: **Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on**. [S.l.: s.n.], 1999. v. 3, p. –1640 Vol. 3.

VASCO, D. W.; JR, J. E. P.; MAJER, E. L. Nonuniqueness in travelttime tomography: Ensemble inference and cluster analysis. **Geophysics**, Society of Exploration Geophysicists, v. 61, n. 4, p. 1209–1227, 1996.

WHITLEY, D. et al. Evaluating evolutionary algorithms. **Artificial Intelligence**, v. 85, p. 245–276, 1996.

ZHANG, J.; ZHANG, J.-R.; LI, K. A sequential niching technique for particle swarm optimization. In: HUANG, D.-S.; ZHANG, X.-P.; HUANG, G.-B. (Ed.). **Advances in Intelligent Computing**. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3644). p. 390–399. ISBN 978-3-540-28226-6.