

# ESTUDO PARAMÉTRICO DO FSS II

Trabalho de Conclusão de Curso  
Engenharia de Computação

Aluno: Luiz Felipe Vieira Verçosa

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho

**Luiz Felipe Vieira Verçosa**

## ***Estudo Paramétrico do FSS II***

Monografia apresentada para obtenção do  
Grau de Bacharel em Engenharia de Com-  
putação pela Universidade de Pernambuco

Orientador:

Prof. Dr. Carmelo José Albanez Bastos Filho

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO  
ESCOLA POLITÉCNICA DE PERNAMBUCO  
UNIVERSIDADE DE PERNAMBUCO

Recife - PE, Brasil

Dezembro de 2014

Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação  
Projeto de Final de Curso



### MONOGRAFIA DE FINAL DE CURSO

#### Avaliação Final (para o presidente da banca)\*

No dia 9 de 12 de 2014, às 16:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente LUIZ FELIPE VIEIRA VERCOSA, orientado pelo professor Carmelo José Albanéz Bastos Filho, sob título Estudo Paramétrico do Algoritmo de Busca por Cardumes Versão dois, a banca composta pelos professores:

**Fernando Buarque de Lima Neto**

**Carmelo José Albanéz Bastos Filho**

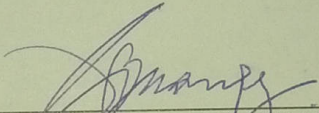
Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

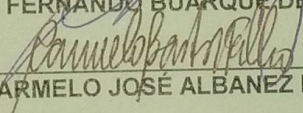
Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 9,0 ( *Nove* )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

  
\_\_\_\_\_  
FERNANDO BUARQUE DE LIMA NETO

  
\_\_\_\_\_  
CARMELO JOSÉ ALBANEZ BASTOS FILHO

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

## Resumo

Inteligência de Enxames (*SI, Swarm Intelligence*) é definido como uma propriedade de qualquer sistema em que a interação entre indivíduos com comportamento simples em uma população, pode resolver problemas complexos. A Otimização Baseada em Cardumes (*FSS, Fish School Search*) é um algoritmo de Inteligência de Enxames com bom desempenho em funções multimodais e inspirado nos mecanismos de alimentação e movimentação coordenada presente em cardumes naturais. Já o Algoritmo de Otimização por Cardumes Versão II (*FSS II, Fish School Search II*) é uma melhoria do FSS e foi criado com o objetivo de superar limitações presentes na primeira versão, como a necessidade de avaliar o *fitness* duas vezes por peixe por iteração e de regular parâmetros de acordo com a função de teste. Entretanto, o FSS II apresenta problemas de instabilidade que dificultam sua convergência em alguns casos.

Neste trabalho de conclusão de curso, é realizado um estudo dos parâmetros do FSS II seguido de modificações que melhoram o seu desempenho. A versão modificada do algoritmo é denominada de FSS II Modificado e as melhorias são feitas por meio de mudanças nos componentes volitivo e instintivo, conservando propriedades importantes do FSS II com a redução de complexidade do algoritmo com relação à primeira versão. O componente volitivo é melhorado por meio da reintrodução do passo volitivo, presente no FSS, enquanto que o termo instintivo passa a usar a variação do peso dos peixes ao invés dos valores absolutos dos seus pesos.

Para um estudo comparativo, foram utilizados os algoritmos FSS, FSS II Original, FSS II Modificado e PSO Global, os quais foram testados com funções de teste amplamente utilizadas na comunidade científica para otimização em larga escala: Esfera, Rastrigin, Rosenbrock, Griewank e Ackley. Os resultados obtidos mostram que o FSS II Modificado supera o desempenho do FSS II Original em todas as funções de teste e é competitivo com o FSS.

## *Abstract*

Swarm Intelligence is defined as a property from any systems where the interaction among its simple individuals create behavior capable of solving complex problems. Fish School Search is a swarm-based optimizer that excels on multimodal search problems. The Fish School Search Version Two has been created in order to overcome some drawbacks from the first version such as the need to evaluate the fitness function twice per fish per iteration and the necessity to proper define the step used in some operators. However, FSS II has showed some instability that set back the algorithm convergence in some cases.

In this monograph, we present a FSS II parameter study followed by modifications that improve the algorithm performance. The modified version is called Modified FSS II. The improvements are carried out in the volitive and instinctive components preserving important FSS II's complexity reduction when compared to FSS. The volitive component is improved by employing the volitive step from FSS while the instinctive component uses the fish weight variation instead of the absolute values of the weight.

We selected the algorithms FSS, Original FSS II, Modified FSS II and Global PSO for a comparative study. In addition, scientific widely known benchmark functions for optimization in large scale such as Sphere, Rastrigin, Rosenbrock, Griewank and Ackley has been used for testing. The results show that Modified FSS II outperforms Original FSS II in all tested functions and is competitive with FSS.

## *Dedicatória*

Dedico este trabalho à memória de meu avô, José Paulo de Oliveira Verçosa e a do meu tio Marcelo Verçosa.

## *Agradecimentos*

Primeiramente agradeço aos meus pais e minha irmã, por serem uma família espetacular e meus melhores amigos. Meus pais, por terem me dado uma ótima educação desde a infância, tanto no aspecto intelectual quanto moral. Eles foram meu braço de apoio quando tropecei e as palavras de esperança quando descreditei. Sem eles não sei até onde teria chegado, mas com eles sei que posso ir muito longe. A minha irmã agradeço pela cumplicidade, conselhos e bom humor. Fabíola é uma das pessoas mais inteligentes que conheço e sei que somos muito melhores juntos.

A todos os meus primos, primas, tios, tias, avós e avôs tanto os que se foram como os que continuam conosco. Vocês são pessoas que eu respeito e admiro muito. Agradeço especialmente à 'Vovó', Maria José de Oliveira Verçosa, por ser uma avó simplesmente incrível. Apesar de todos os empecilhos de sua vida, sempre foi uma pessoa forte, honesta e boa. Qualquer dia ou hora que cheguei em sua casa sempre tive o melhor macarrão com bife do mundo! 'Vó', a senhora é minha inspiração e minha força!

Ao Professor Carmelo por ter me orientado com excelência durante esse trabalho de conclusão de curso. É possível notar em suas palavras, sabedoria e paixão pelo que faz. Também agradeço a professora Eliane Maria Loiola pela paciência e orientação durante a minha iniciação científica. E aos demais professores incríveis como Alberto do SENAI, Edgar do Colégio Contato e todos que passaram e marcaram minha vida no período de estudos no Colégio Souza Leão, Colégio Contato, Escola Técnica SENAI de Areias e é claro Universidade de Pernambuco, não só ensinando o conteúdo da disciplina, mas principalmente formando cidadãos. Vocês são a essência do que a sociedade brasileira precisa para progredir!

Por fim e não menos importante, agradeço a todos os amigos da POLI, especialmente meus companheiros de turma. Tive simplesmente o privilégio de estudar em uma turma muito unida e muito bem-humorada. Aprendi que o bom-humor é capaz de dar um sentido todo especial a vida, nos propiciando momentos felizes e permitindo que possamos responder melhor às adversidades. Galera, eu não teria passado

em Arquitetura de Computadores se não fosse por vocês! Meus amigos, "tiro meu chapéu"pra vocês! Meu abraço e muito obrigado!



*“Na vida, você precisa de três coisas:  
Jesus no coração,  
saúde pra trabalhar  
e dinheiro no bolso”*  
– José Paulo de O. Verçosa, meu avô

# *Sumário*

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Algoritmos</b>	<b>xi</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Caracterização do Problema . . . . .	1
1.2 Hipóteses e Objetivos . . . . .	3
1.3 Organização do Documento . . . . .	3
<b>2 Inteligência de Enxames</b>	<b>4</b>
2.1 Inteligência de Enxames . . . . .	4
2.2 Otimização por cardume . . . . .	5
2.2.1 Otimização por Cardume Versão Original (FSS) . . . . .	6
2.2.2 Otimização por Cardume Versão 2 (FSS II) . . . . .	10
<b>3 Contribuição</b>	<b>15</b>
3.1 Problemas da Versão Original do FSS II . . . . .	15
3.1.1 O Problema do Termo Volitivo . . . . .	15
3.2 Contribuição: FSS II Modificado . . . . .	17
3.2.1 Análise da Influência dos Operadores . . . . .	18

<b>4</b>	<b>Resultados e Discussão</b>	<b>23</b>
4.1	Arranjo Experimental . . . . .	23
4.2	Parâmetros utilizados . . . . .	24
4.2.1	FSS II Original . . . . .	25
4.2.2	FSS II Modificado . . . . .	27
4.2.3	FSS . . . . .	27
4.2.4	PSO-G . . . . .	27
4.3	Simulações . . . . .	28
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>36</b>
5.1	Conclusões . . . . .	36
5.2	Trabalhos Futuros . . . . .	37
	<b>Referências Bibliográficas</b>	<b>39</b>

## *Lista de Figuras*

2.1	Ilustração do movimento coletivo-instintivo antes (cinza escuro) e após (cinza claro) sua ocorrência. Extraída de [1]. . . . .	8
2.2	Ilustração do movimento coletivo-volitivo antes (cinza escuro) e após (cinza claro) sua ocorrência. Extraída de [1]. . . . .	10
3.1	Peixes estagnados em suas posições iniciais. Imagem obtida utilizando-se o projeto Doss criado pelo Grupo de Pesquisa em Inteligência de Enxames - POLI. . . . .	16
3.2	Perda de diversidade do cardume na dimensão x. Imagem obtida utilizando-se o projeto Doss criado pelo Grupo de Pesquisa em Inteligência de Enxames - POLI. . . . .	17
3.3	Vetor de movimento individual do FSS II Original ao longo das iterações de busca na função Esfera. . . . .	19
3.4	Vetor de movimento instintivo do FSS II Original ao longo das iterações de busca na função Esfera. . . . .	19
3.5	Vetor de movimento volitivo do FSS II Original ao longo das iterações de busca na função Esfera. . . . .	20
3.6	Vetores de movimento do FSS II Modificado ao longo das iterações de busca na função Esfera. . . . .	21
3.7	Vetores de movimento do FSS ao longo das iterações de busca na função Esfera. . . . .	21
3.8	Vetores de movimento do PSO-Global ao longo das iterações de busca na função Esfera . . . . .	22
4.1	Representação Gráfica da Função Esfera em duas dimensões. . . . .	25
4.2	Representação Gráfica da Função Rastrigin em duas dimensões. . . . .	25
4.3	Representação Gráfica da Função Rosenbrock em duas dimensões. . . . .	26

4.4	Representação Gráfica da Função Ackley em duas dimensões. . . . .	26
4.5	Evolução do valor de melhor <i>fitness</i> para a função Esfera em dez mil iterações e trinta dimensões. . . . .	29
4.6	Evolução do valor de melhor <i>fitness</i> para a função Rastrigin em trinta dimensões e dez mil iterações. . . . .	30
4.7	Evolução do valor de melhor <i>fitness</i> para a função Rosenbrock em trinta dimensões e dez mil iterações. . . . .	30
4.8	Evolução do valor de melhor <i>fitness</i> para a função Griewank em trinta dimensões e dez mil iterações. . . . .	31
4.9	Evolução do valor de melhor <i>fitness</i> para a função Ackley em trinta dimensões e dez mil iterações. . . . .	31
4.10	Boxplot dos resultados de <i>fitness</i> para a Função Esfera em trinta dimensões e dez mil iterações. . . . .	32
4.11	Boxplot dos resultados de <i>fitness</i> para a Função Rastrigin em trinta dimensões e dez mil iterações. . . . .	33
4.12	Boxplot dos resultados de <i>fitness</i> para a Função Rosenbrock em em trinta dimensões e dez mil iterações. . . . .	33
4.13	Boxplot dos resultados de <i>fitness</i> para a Função Griewank em em trinta dimensões e dez mil iterações. . . . .	34
4.14	Boxplot dos resultados de <i>fitness</i> para a Função Ackley em em trinta dimensões e dez mil iterações. . . . .	34

## *Lista de Tabelas*

4.1	Espaço de busca, de inicialização e ponto ótimo das funções de teste utilizadas nas simulações. . . . .	27
4.2	Configuração dos passos volitivo ( $s_{vol}$ ) e individual ( $s_{ind}$ ) para simulação do FSS. . . . .	27
4.3	Valores de melhor fitness e desvio padrão correspondente. O desvio padrão de cada função está entre parêntesis. . . . .	28

## *Lista de Algoritmos*

1	Pseudocódigo do FSS. . . . .	10
2	Pseudocódigo do FSS II. . . . .	14

## *Lista de Abreviaturas e Siglas*

**ABC** – *Artificial Bee Colony*

**ACO** – *Ant Colony Optimization*

**FSS** – *Fish School Search*

**FSS II** – *Fish School Search Version 2*

**PBA** – *Population Based Optimization Algorithms*

**PSO** – *Particle Swarm Optimization*

**SI** – *Swarm Intelligence*



# 1 *Introdução*

*“Fazer ou não fazer,  
não existe tentar.”*

**– Mestre Yoda.**

Neste trabalho de conclusão de curso foram desenvolvidas melhorias na segunda versão do Algoritmo de Busca por Cardumes também chamada de FSS II (*Fish School Search II*). A escolha da proposta deveu-se ao fato de que o FSS II é um algoritmo recente, promissor e que precisa de ajustes para melhorar seu comportamento.

Este capítulo apresenta a introdução desta monografia, e está organizado em 3 Seções. Na Seção (1.1), são apresentadas as características dos problemas otimizados pelos algoritmos de enxames assim como exemplos de criação e desenvolvimento desses algoritmos. Em seguida, na Seção (1.2) são criadas hipóteses de como será possível aperfeiçoar o FSS II. Por fim, na Seção (1.3), a estrutura do restante da monografia é descrita.

## 1.1 **Motivação e Caracterização do Problema**

Problema de otimização se caracteriza pela busca da melhor solução entre todas as soluções possíveis para um determinado problema. Para problemas simples, com duas variáveis de entrada por exemplo, pode-se encontrar a melhor solução utilizando os conceitos matemáticos de diferenciação e integração. Entretanto, existem problemas mais complexos que podem possuir dezenas de variáveis de entrada, combinações não lineares entre elas e ainda espaços de busca amplos. Considere um problema hipotético em que seja necessário avaliar variáveis como: temperatura, umidade, pressão, velocidade do vento, luminosidade, altura e acidez do solo para realizar um obra de engenharia, por exemplo. Para tal problema pode ser inviável utilizar métodos matemáticos tradicionais. Felizmente, pode-se fazer uso de algoritmos de

otimização que apesar de não garantirem sempre a melhor solução, ao menos podem garantir soluções aceitáveis [2], rapidamente.

Algoritmos de otimização são técnicas para busca por uma solução ótima para um certo problema que pode ser modelado por funções objetivo. Inspirações oriundas da natureza permitiram o desenvolvimento de modelos capazes de resolver diversos problemas complexos de otimização. Esses modelos fazem parte do campo de estudo chamado Inteligência Computacional [2].

A Inteligência Computacional provê diversos algoritmos baseados em população inspirados na natureza ou PBA (*Population Based Optimization Algorithms*). Essas técnicas utilizam um conjunto de entidades para coletivamente e coordenadamente realizar uma busca. Inteligência de Enxames, SI (*Swarm Intelligence*), é definido como uma propriedade de qualquer sistema no qual a interação entre indivíduos com comportamento simples em uma população, gera padrões complexos [3]. Como exemplos, tem-se o Algoritmo de Otimização por Enxame de Partículas ou PSO (*Particle Swarm Optimization*), que é uma técnica SI baseada no comportamento de um bando de pássaros [4]; o Algoritmo de Otimização por Colônia de Formigas, o ACO (*Ant Colony Optimization*) [5], que baseia-se na busca das formigas por alimento; e Algoritmo de Busca por Cardume, FSS (*Fish School Search*) [1], que se inspira na busca de alimento em uma população de peixes.

O Algoritmo de Otimização por Cardume foi proposto por Bastos Filho e Lima Neto em 2008 [1] em sua versão original. Apesar de inovador e de apresentar resultados interessantes para problemas multimodais em alta dimensionalidade, o FSS possui limitações relacionadas ao custo computacional e à necessidade de parâmetros específicos. Por isso, em 2013, uma versão aprimorada do algoritmo foi proposta por Bastos Filho e Nascimento [10]. Mesmo introduzindo conceitos interessantes e removendo as restrições mencionadas da versão anterior, o Algoritmo de Otimização por Cardume Versão Dois, o FSS II, (*Fish School Search II*) apresenta instabilidades para convergência em funções de teste conhecidas, como Esfera, Rastrigin, Rosenbrock, Griewank e Ackley. Faz-se necessário, um estudo paramétrico do FSS II avaliando também a influência dos operadores para identificar os problemas e propor soluções.

## 1.2 Hipóteses e Objetivos

Uma vez que o FSS II possui conceitos interessantes como menor custo computacional e eliminação de parâmetros específicos presentes na versão original, mas possui algumas limitações de convergência, é desejável tornar o algoritmo mais eficiente. Assim, hipotetiza-se que o FSS II não converge bem porque precisa de ajustes em seus parâmetros e operadores. Dado que o FSS II é uma versão promissora do FSS, é portanto razoável, mais estudos e modificações em seus parâmetros e operadores a fim de alcançar seu potencial de otimização inicialmente antecipado.

O objetivo deste trabalho é assim, estudar e modificar os parâmetros e operadores do FSS II a fim de obter uma versão mais estável, com melhor convergência e maior capacidade de otimizar funções.

## 1.3 Organização do Documento

Este trabalho está organizado em 5 capítulos. No Capítulo 2 é abordado o conceito de Inteligência de Enxames e descrições detalhadas do FSS e FSS II. Em seguida, no Capítulo 3, é apresentada a contribuição deste trabalho de conclusão de curso: o FSS II Modificado que é uma versão melhorada do FSS II. No Capítulo 5, os experimentos e resultados serão apresentados. Por fim, no Capítulo 6, são discutidas as principais conclusões desse trabalho, como também propostas para trabalhos futuros.

## 2 *Inteligência de Enxames*

*“Não importa o quanto você bate, mas sim  
o quanto aguenta apanhar e continuar.  
O quanto pode suportar e seguir em frente.  
É assim que se ganha.”*

**– Rocky Balboa.**

Neste capítulo é apresentada a base necessária à compreensão da proposta desse trabalho. Na Seção (2.1) é apresentada uma breve abordagem sobre inteligência de enxames e na Seção (2.2) é mostrada uma detalhada descrição das duas versões dos algoritmos de busca por cardume e suas diferenças.

### 2.1 **Inteligência de Enxames**

Inteligência de enxames (*SI, Swarm Intelligence*) é uma classe de algoritmos pertencentes à área de Inteligência Computacional que são inspirados em populações naturais de enxames. Inicialmente introduzidos em conferências de Computação Evolucionária [6], esses algoritmos tem sido aplicados com sucesso em vários problemas reais de engenharia e outras áreas [7] [8] [9], sendo usados principalmente em problemas de otimização.

Em SI, modelos computacionais de insetos e outros animais possuem memória limitada e são geralmente chamados de entidades. Essas entidades trocam informações e por meio dessa troca, uma inteligência coletiva emerge, em que cada entidade possui informação global ou local sobre o espaço de busca. Baseados nessas informações, as entidades movem-se para regiões que são mais propícias de possuírem boas soluções para o problema de otimização. Uma vez que o comportamento de cada entidade é simples, os algoritmos de SI geralmente são computacionalmente eficientes, superando na maioria das vezes, métodos tradicionais de otimização.

Diferentes algoritmos SI são inspirados em diferentes animais. Por exemplo, o algoritmo de otimização por enxame de partículas (*PSO, Particle Swarm Optimization*) proposto em 1995 pelos pesquisadores Kennedy e Eberhart [4] baseia-se no comportamento de um bando de pássaros em busca de comida. Notou-se que os pássaros voavam em certa sincronia mesmo com mudanças repentinas de direção quando em busca por comida. Com essa e outras observações, criou-se um modelo simplificado para resolver problemas contínuos onde cada pássaro foi representado por uma partícula de cuja interação com outras partículas, emergia inteligência.

Outro exemplo bastante popular é o algoritmo de otimização por colônia de formigas (*ACO, Ant Colony Optimization*) [5], que baseia-se na busca das formigas por alimento. Nos estudos realizados notou-se que ao andar, cada formiga deposita naturalmente uma substância química chamada ferormônio a qual atrai outras formigas. Concluiu-se que quanto maior a quantidade de ferormônio em um caminho, maior a probabilidade de ele ser promissor em alimentos e de atrair mais formigas. Com isso, criou-se um modelo simplificado para resolver problemas combinatoriais discretos.

Existem, de fato, vários algoritmos inteligentes inspirados no comportamento de diferentes animais e que são úteis para solucionar diversas classes de problemas. Deve-se salientar, portanto, que dado um problema, um algoritmo apropriado deve ser selecionado para resolvê-lo, e não o contrário.

O Algoritmo de Otimização por Cardume (*FSS, Fish School Search*) [1] inspira-se na busca de alimento por parte de uma população de peixes. Ele é um algoritmo recente quando comparado ao ano de surgimento da maioria dos outros. Por exemplo, o PSO surgiu em 1995, enquanto que a primeira versão do FSS foi lançada em 2008. Naturalmente, algoritmos recentes precisam de reajustes e melhorias. Esse é um dos principais motivos do aprofundamento do estudo do FSS por meio desse trabalho.

## 2.2 Otimização por cardume

O algoritmo de otimização baseado em cardumes foi proposto em 2008 [1] e, como já mencionado, baseia-se na busca de alimentos por parte de um cardume. Foram observados os seguintes aspectos presentes em cardumes reais:

1. Os peixes se organizam em cardumes de maneira a evitar o ataque de predadores e facilitar a obtenção de comida;

2. Cada peixe possui certa independência para buscar alimento por conta própria sem por isso abandonar o cardume;
3. O cardume também nada para uma mesma direção no oceano;
4. Quando a comida torna-se abundante em um certo ponto do cardume, ele contrai-se em torno daquela região para alimentar-se. Em contrapartida, quando a comida torna-se esparsa, ocorre uma dilatação no cardume com o intuito de aumentar a região de busca;

Em uma adaptação do cardume real para o algoritmo, cada peixe torna-se uma entidade e a busca por comida corresponde à busca por melhores soluções (ou *fitness*) na função a ser otimizada. Em outras palavras, os peixes movem-se em um grande aquário de  $n$  dimensões onde cada dimensão representa uma variável a ser otimizada e onde os limites são os valores máximo e mínimo que cada variável da função pode assumir.

Com base nesses aspectos, desenvolveu-se a primeira versão do FSS [1]. Posteriormente, o algoritmo foi aperfeiçoado originando a segunda versão chamada de FSS II [10] que é derivado diretamente da primeira. Há também uma versão do FSS denominada *Weight-Based Fish School Search* ou wFSS [11], a qual utiliza-se de múltiplos cardumes para melhorar a otimização multimodal. Porém, essa última versão não encontra-se no escopo desse trabalho o qual foca-se na melhoria do FSS II e na comparação com a versão que deu origem ao mesmo, o FSS. A seguir, tanto o FSS, como o FSS II, são detalhados.

### 2.2.1 Otimização por Cardume Versão Original (FSS)

A primeira versão do algoritmo de busca por cardumes também é chamada de FSS. Antes de descrever o comportamento dos peixes no algoritmo, faz-se necessário definir alguns conceitos intrínsecos à busca por cardume. O primeiro deles é o peso individual dos peixes. Esse peso é a medida de quão gordo (e bem-sucedido) é o peixe. Peixes gordos representam um histórico de sucesso e exercem uma maior influência no cardume que os magros.

O segundo conceito é o de passo. O passo, como o nome sugere, representa quanto o peixe irá nadar em certa dimensão. Sendo assim, ele representa um deslocamento no espaço de busca em certa dimensão e controla a busca por profundidade

e por largura do algoritmo.

O terceiro e último conceito é a divisão do movimento do cardume em duas categorias, que são: movimento individual e movimento coletivo. Cada movimento possui uma correspondência direta com as características observadas no cardume real e listadas na Seção (2.2).

### Movimento Individual

O movimento individual corresponde ao item 2 da Seção (2.2) e é dado pela Equação (2.1):

$$\vec{n}(t+1) = \vec{x}(t) + s_{ind}(t+1) \cdot \vec{U}[-1; 1], \quad (2.1)$$

em que  $s_{ind}(t+1)$  corresponde ao passo do movimento individual e  $\vec{U}[-1; 1]$  é um vetor formado por uma função de distribuição de probabilidade uniforme que gera valores aleatórios no intervalo entre -1 e 1.

O movimento individual é guloso, ou seja, o peixe moverá-se apenas se o valor de *fitness* na nova posição for melhor que o da posição anterior, ou seja, se  $\Delta f_i(t+1) > 0$ . Além disso, a nova posição gerada é aleatória e vizinha à anterior. A variação do *fitness* é calculada pela Equação (2.2).

$$\Delta f_i(t+1) = f[\vec{n}_i(t+1)] - f[\vec{x}_i(t)]. \quad (2.2)$$

É importante observar também que cada peixe precisará avaliar o *fitness* tanto da posição vizinha ( $\vec{n}(t+1)$ ) quanto da posição atual ( $\vec{x}(t)$ ) para poder decidir se, de fato, irá ou não mover-se.

### Movimento Coletivo Instintivo

O movimento coletivo instintivo do cardume é um deslocamento vetorial de todo o cardume para uma mesma direção no aquário (espaço de busca). Esse movimento vetorial é influenciado apenas por peixes que obtiveram sucesso no movimento individual e é calculado usando a Equação 2.3:

$$\vec{m}(t+1) = \frac{\sum_{i=1}^n (\Delta \vec{x}_i(t+1) \Delta f_i(t+1))}{\sum_{i=1}^n \Delta f_i(t+1)}, \quad (2.3)$$

$$\Delta \vec{x}_i(t+1) = \vec{n}(t+1) - \vec{x}(t), \quad (2.4)$$

em que  $\vec{m}(t+1)$  corresponde ao vetor instintivo;  $\Delta \vec{x}_i(t+1)$  equivale à variação na posição do peixe  $i$  no movimento individual e  $\Delta f_i(t+1)$  é a variação do *fitness* do peixe  $i$  também com base no movimento individual. É importante lembrar que só os peixes que obtiveram melhor *fitness* no movimento individual participam do movimento instintivo, conseqüentemente, todas as variações de *fitness* presentes na Equação (2.3) são positivas.

Após o cálculo do vetor, as posições dos peixes devem ser atualizadas com:

$$\vec{x}(t+1) = \vec{x}(t) + \vec{m}(t+1). \quad (2.5)$$

A Figura (2.1) ilustra o movimento instintivo.

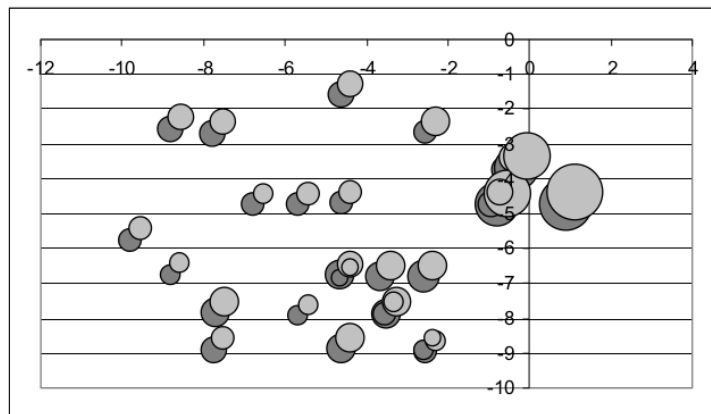


Figura 2.1: Ilustração do movimento coletivo-instintivo antes (cinza escuro) e após (cinza claro) sua ocorrência. Extraída de [1].

### Movimento Coletivo Volitivo

O movimento coletivo volitivo do cardume ocorre na forma de uma contração ou dilatação do cardume no aquário e corresponde ao item 4 da Seção (2.2). Ele ocorre na forma de afastamento ou aproximação do centro de gravidade (baricentro) do cardume, o qual é calculado com a Equação (2.6).

$$\vec{B}(t+1) = \sum_{i=1}^n \frac{\vec{x}_i(t)W_i(t)}{\vec{x}_i(t)}, \quad (2.6)$$



em que  $\vec{x}_i(t)$  representa a posição do peixe  $i$  na iteração  $t$  e  $W_i(t)$  é o peso do peixe  $i$  também na iteração  $t$ . O peso, por sua vez, é atualizado de acordo com a Equação (2.7).

$$W_i(t+1) = W_i(t) + \frac{\Delta f_i(t+1)}{\max[|\Delta f(t+1)|]}, \quad (2.7)$$

em que  $\Delta f_i(t+1)$  é a variação do *fitness* do peixe  $i$  na iteração  $t+1$  e  $\max[|\Delta f(t+1)|]$  é a maior variação em módulo do *fitness* entre todos os peixes na iteração  $t+1$ . Nota-se que o ganho de peso é normalizado, o que impede uma grande oscilação nos pesos dos peixes.

Por fim, a Equação que define o movimento volitivo é dada por (2.8).

$$\vec{x}(t+1) = \vec{x}(t) - \text{sign}\left[\sum_{i=1}^n \Delta W_i(t+1)\right] \cdot s_{vol}(t+1) \cdot \vec{U}[0;1] \cdot \frac{\vec{x}_i(t+1) - \vec{B}(t+1)}{\text{dist}[x_i(t+1), B(t+1)]}, \quad (2.8)$$

em que  $\Delta W_i(t+1)$  equivale à variação de peso do peixe  $i$  entre as duas últimas iterações;  $\text{sign}[\cdot]$  é uma função que retorna o sinal do argumento;  $s_{vol}(t+1)$  corresponde ao passo do movimento volitivo;  $\vec{U}[0;1]$  é um vetor formado por uma função de distribuição de probabilidade uniforme que gera valores aleatórios no intervalo entre 0 e 1 e  $\text{dist}[\cdot]$  é uma função que retorna a distância euclidiana entre o baricentro e a posição atual do peixe. Em outras palavras, considerando que o movimento é vetorial,  $\text{sign}[\cdot]$  retornará 1 se o cardume engordou determinando que o sentido do movimento seja em direção ao baricentro. Caso o cardume haja emagrecido, a função  $\text{sign}[\cdot]$  retornará -1 determinando que o sentido do movimento seja em direção contrária ao baricentro. O parâmetro  $s_{vol}$  indica a magnitude do movimento, e por fim o termo  $\frac{\vec{x}_i(t+1) - \vec{B}(t+1)}{\text{dist}[x_i(t+1), B(t+1)]}$  determinará a direção do movimento.

A Figura (2.2) ilustra a execução desse movimento em um caso onde ocorre uma contração do cardume.

### Pseudocódigo do FSS

O Algoritmo (1) descreve o pseudocódigo do FSS.

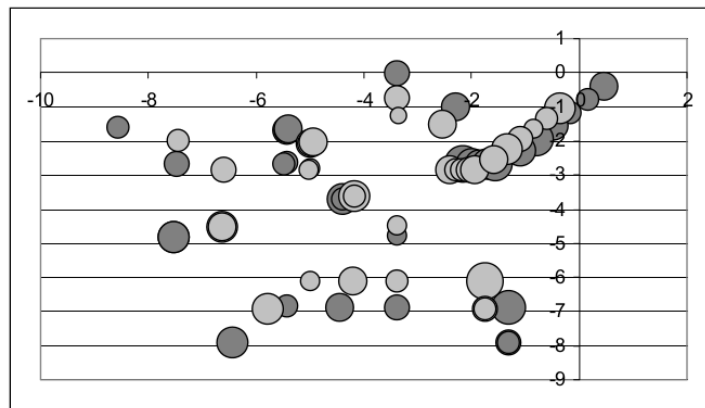


Figura 2.2: Ilustração do movimento coletivo-volitivo antes (cinza escuro) e após (cinza claro) sua ocorrência. Extraída de [1].

---

**Algoritmo 1:** Pseudocódigo do FSS.

---

1 **início**

2 Inicialize todos os peixes em posições  $\vec{x}_i(0)$  aleatórias e distantes do mínimo global;

3 Inicialize aleatoriamente o peso  $\vec{w}_i(0)$  de todos os peixes;

4 **enquanto** *critério de parada não for alcançado* **faça**

5     **para cada** *peixe* **faça**

6         Encontre posição vizinha (2.1);

7         Avalie a variação no fitness de acordo com (2.2) e mova o peixe apenas se o *fitness* melhorou;

8         Alimente os peixes utilizando (2.7);

9     Calcule o vetor do movimento instintivo utilizando (2.3);

10     **para cada** *peixe* **faça**

11         Execute o movimento instintivo utilizando (2.5);

12     Calcule o baricentro utilizando (2.6);

13     **para cada** *peixe* **faça**

14         Execute o movimento volitivo usando (2.8);

15     Atualize  $s_{ind}$  e  $s_{vol}$ .

---

### 2.2.2 Otimização por Cardume Versão 2 (FSS II)

A segunda versão do algoritmo de busca por cardumes surgiu em 2013 com o intuito de aprimorar a convergência do FSS e diminuir seu custo computacional. Os

principais aspectos de melhoria do FSS II foram:

1. Retirar os passos individual e volitivo. O FSS II usa um mecanismo de auto-adaptação (será mais explicado adiante) e não precisa de passos. A vantagem dessa abordagem é que ela confere mais independência ao cardume. O passo, ao contrário, controla demasiadamente a granularidade da busca, pois se é grande, dificulta a busca em profundidade e se é pequeno, torna a convergência do algoritmo muito lenta;
2. O FSS II só precisa de uma avaliação de fitness por peixe por iteração, enquanto que o FSS precisa de duas avaliações de *fitness*;
3. A atualização de posição de cada peixe é feita de forma mais rápida, pois os diferentes movimentos são adicionados em uma única equação. Isso é possível devido ao fato de que no FSS II os peixes sempre se movem em todos os tipos de movimento enquanto que no FSS, nem todos os peixes se movem no movimento individual;

Como mencionado anteriormente, o FSS II atualiza a posição do cardume em apenas um laço utilizando a Equação (2.9), para cada peixe:

$$\begin{aligned} \vec{x}_i(t+1) = \vec{x}_i(t) + \beta \cdot c \cdot \Delta \vec{x}_i(t+1) + c \cdot \vec{U}(0,1) \cdot \frac{\sum_{i=1}^n (\Delta \vec{x}_i(t+1) W_i(t+1))}{\sum_{i=1}^n W_i(t+1)} + \\ c \cdot \vec{U}(0,1) \cdot \text{sign} \left[ \sum_{i=1}^n \Delta W_i(t+1) \right] \cdot [\vec{x}_i(t) - \vec{B}(t+1)], \end{aligned} \quad (2.9)$$

em que  $\vec{x}_i(t+1)$  é a nova posição do peixe;  $\vec{x}_i(t)$  é sua posição anterior;  $\beta$  é um parâmetro estático;  $c$  é o parâmetro dinâmico auto-adaptativo (será aprofundado mais a frente);  $\Delta \vec{x}_i(t+1)$  é a variação na posição do peixe  $i$  nas duas últimas iterações;  $W_i(t+1)$  é o peso atual do peixe  $i$ ;  $\text{sign}[\cdot]$  é uma função que retorna o sinal do argumento e  $\vec{B}(t+1)$  é a posição do baricentro. Posteriormente, será mostrado quais valores os parâmetros assumem nos testes. As Equações referentes à variação de peso e *fitness* são mostradas pelas Equações (2.10), (2.11) e (2.12):

$$\Delta f_i(t+1) = f[\vec{x}_i(t)] - f[\vec{x}_i(t-1)], \quad (2.10)$$

$$W_i(t+1) = W_i(t) + \Delta f_i(t+1), \quad (2.11)$$

$$\Delta W_i(t+1) = W_i(t+1) - W_i(t). \quad (2.12)$$

Nota-se que em (2.9) os diferentes vetores de movimento estão reunidos em uma só Equação. Nas seções subsequentes, diferentes componentes da Equação (2.9) serão elucidados individualmente para facilitar o entendimento e comparação com a versão anterior do algoritmo.

### Parâmetro $c$

A auto-adaptação do FSS II é gerada pelo parâmetro dinâmico  $c$ . Esse parâmetro é guiado pelo parâmetro estático  $\alpha$  e se ajusta automaticamente em tamanho dependendo do sucesso momentâneo da busca. Para tanto, o  $c$  considera o número de peixes que ganharam peso em cada iteração usando a seguinte regra: inicia-se a busca atualizando o valor de  $c$  com a fórmula:  $c = c \cdot (1 - \alpha)$ . Se o número de peixes que aumentaram em peso for menor que na iteração anterior, o sinal de  $\alpha$  é invertido, senão, o sinal é mantido. Por exemplo, se o valor de  $c$  está sendo atualizado com a fórmula:  $c = c \cdot (1 - \alpha)$  e o número de peixes que engordaram diminuiu,  $c$  passará a ser atualizado com a fórmula  $c = c \cdot (1 + \alpha)$ . No entanto, se  $c$  está sendo atualizado com a fórmula:  $c = c \cdot (1 - \alpha)$  e o número de peixes que engordaram aumentou, então  $c$  continuará a ser atualizado com exatamente a mesma fórmula:  $c = c \cdot (1 - \alpha)$ .

### Termo que Substitui o Movimento Individual

O movimento individual do FSS original é modificado no FSS II para se tornar um termo que gera inércia, como pode ser visto na Equação (2.13):

$$\beta \cdot c \cdot \Delta \vec{x}_i(t+1), \quad (2.13)$$

em que  $\beta$  é um parâmetro de inércia;  $c$  é o parâmetro dinâmico auto-adaptativo e  $\Delta \vec{x}_i(t+1)$  é a variação na posição do peixe  $i$  nas duas últimas iterações dada por:

$$\Delta \vec{x}_i(t+1) = [\vec{x}_i(t) - \vec{x}_i(t-1)]. \quad (2.14)$$

É importante frisar que a Equação (2.13) é parte de (2.9) e que está sendo trazida individualmente para facilitar seu entendimento e comparação com o FSS.

Há duas diferenças importantes entre os movimentos individuais do FSS e do FSS II. A primeira é que o FSS II não usa o passo individual ( $s_{ind}$ ) e a segunda é que ao contrário do que acontece na versão original, todos os peixes realizam o movimento individual no FSS II. Conseqüentemente, no FSS II o movimento não é guloso sendo desnecessário comparar o *fitness* da posição atual com o da vizinha. A vantagem disso é o custo computacional porque o cálculo do *fitness* é na maioria das vezes dispendioso. Por outro lado, observa-se que esse movimento ganha uma conotação diferente na nova versão do algoritmo de cardumes.

### Movimento Coletivo Instintivo

O movimento instintivo no FSS II também está presente em (2.9) e é representado por:

$$c \cdot \vec{U}(0, 1) \cdot \frac{\sum_{i=1}^n (\Delta \vec{x}_i(t+1) W_i(t+1))}{\sum_{i=1}^n W_i(t+1)}. \quad (2.15)$$

Percebe-se que no FSS II, o movimento individual não influencia diretamente no vetor instintivo. Isso ocorre porque enquanto que no FSS, o  $\Delta \vec{x}_i(t+1)$  da Equação do movimento coletivo instintivo representa o deslocamento obtido pelo movimento individual da iteração atual (calculado por (2.4)), no FSS II  $\Delta \vec{x}_i(t+1)$  refere-se ao deslocamento total do peixe entre as duas últimas iterações (calculado por (2.14)). Além disso, o FSS utiliza a variação do *fitness* (dada por (2.2)) no movimento instintivo ao passo que o FSS II utiliza o peso do peixe no mesmo movimento. A vantagem de se considerar o peso ao invés da variação de *fitness*, é que o peso leva em consideração um histórico de sucesso do peixe, enquanto que a melhora no *fitness* representa um sucesso momentâneo.

### Movimento Coletivo Volitivo

O movimento instintivo no FSS II, presente em (2.9), é representado por:

$$c \cdot \vec{U}(0, 1) \cdot \text{sign} \left[ \sum_{i=1}^n \Delta W_i(t+1) \right] \cdot [\vec{x}_i(t) - \vec{B}(t+1)], \quad (2.16)$$

$$\vec{B}(t+1) = \frac{\sum_{i=1}^n \vec{x}_i(t) W_i(t+1)}{\sum_{i=1}^n W_i(t+1)}. \quad (2.17)$$

As principais diferenças entre os movimentos volitivos do FSS e o FSS II são duas: a primeira é que o FSS II não usa o passo volitivo; a segunda é que no FSS II o termo  $\vec{x}_i(t) - \vec{B}(t+1)$  não é dividido pela distância euclidiana de  $\vec{x}_i(t)$  ao baricentro. Como consequência disso, no FSS a expressão  $\frac{\vec{x}_i(t+1) - \vec{B}(t+1)}{dist[\vec{x}_i(t+1), \vec{B}(t+1)]}$  é responsável apenas por fornecer a direção do vetor volitivo, enquanto que o módulo fica a cargo do passo volitivo. Já no FSS II, a expressão equivalente  $\vec{x}_i(t) - \vec{B}(t+1)$  é responsável por prover tanto a direção quanto o módulo do vetor volitivo. Esse módulo exerce grande impacto na magnitude da movimentação dos peixes em cada iteração, pois o volitivo é o único termo onde há subtração de posições. Consequentemente, esse módulo também exerce impacto direto nos módulos dos vetores individual e instintivo.

Um detalhe importante do FSS II é que apenas na primeira iteração do algoritmo, deve-se realizar uma busca local usando a Equação (2.1) do FSS, porém movendo todos os peixes independente do *fitness*. Após isso, inicia-se o laço do algoritmo onde as equações anteriormente descritas são utilizadas.

### Pseudocódigo do FSS II

No Algoritmo (2), é apresentado o pseudo-código do FSS II.

---

#### Algoritmo 2: Pseudocódigo do FSS II.

---

**1 início**

2 Inicialize todos os peixes em posições  $\vec{x}_i(0)$  aleatórias e distantes do mínimo global;

3 Inicialize aleatoriamente o peso  $\vec{w}_i(0)$  de todos os peixes;

4 **enquanto** *critério de parada não for alcançado* **faça**

5     **para cada** *peixe* **faça**

6         Calcule o deslocamento do peixe usando (2.14);

7         Calcule a variação de *fitness* usando (2.10);

8         Alimente o cardume usando (2.11);

9         Calcule a variação de peso utilizando (2.12);

10     Calcule o baricentro usando (2.17);

11     **para cada** *peixe* **faça**

12         Atualize a posição do peixe utilizando (2.9).

---

## 3 *Contribuição*

*“Nenhuma grande descoberta foi feita  
jamais sem um palpite ousado”*

– Isaac Newton.

Neste capítulo será abordado a investigação realizada no FSS II, desde uma análise paramétrica até a proposta de melhorias para facilitar a convergência. Na Seção (3.1) serão apresentados os problemas descobertos e na Seção (3.2), as melhorias criadas.

### 3.1 Problemas da Versão Original do FSS II

O principal fato que motivou a investigação do FSS II foi a verificação de instabilidades devido ao processo de auto-adaptação do parâmetro  $c$ , o que impedia a convergência do algoritmo em alguns casos.

#### 3.1.1 O Problema do Termo Volitivo

O termo volitivo (2.16) desempenha um papel extremamente importante no FSS II, pois é o único componente da Equação (2.9) onde há subtração de posições:  $\vec{x}_i(t) - \vec{B}(t+1)$ . Essa subtração provê a magnitude do vetor volitivo, a qual varia de acordo com a proximidade do peixe ao baricentro. Ela também, até certo ponto, controla a amplitude dos componentes instintivo e individual, já que esses últimos dependem diretamente da variação das duas últimas posições do peixe.

Devido à importância do termo volitivo, realizou-se um estudo paramétrico, enfatizando o parâmetro  $c$  da Equação (2.9). Entretanto, o estudo revelou que o algoritmo não é capaz de convergir mesmo com diferentes combinações dos parâmetros

$\alpha$ ,  $\beta$  e  $c$ . Recorreu-se então à visualização do algoritmo utilizando uma ferramenta gráfica. Nessa ferramenta, configurou-se o problema com apenas duas dimensões para que fosse possível visualizar o movimento dos peixes e ajustou-se os parâmetros para as configurações padrão encontradas no artigo [10], que são  $\alpha = 0,01$ ;  $\beta = 0,4$  e  $c_{inicial} = 0,1$ . Observou-se que com a configuração citada, os peixes ficavam praticamente estagnados em suas posições iniciais como pode ser observado na Figura (3.1).

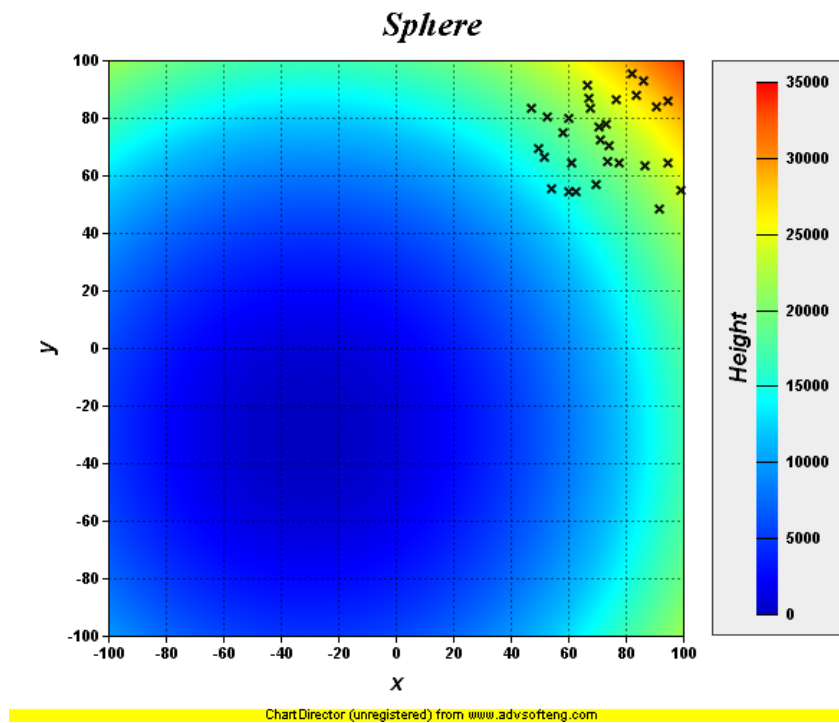


Figura 3.1: Peixes estagnados em suas posições iniciais. Imagem obtida utilizando-se o projeto Doss criado pelo Grupo de Pesquisa em Inteligência de Enxames - POLI.

Após aumentar consideravelmente o valor do parâmetro  $c$ , por exemplo de 0,1 para 0,8, o cardume passava a mover-se pelo espaço de busca, porém perdia frequentemente a diversidade em uma das dimensões como mostrado na Figura (3.2). Notou-se que essa perda de diversidade, devia-se à proximidade da posição do peixe ao baricentro na dimensão em questão. Em outras palavras, uma vez que o peixe aproxima-se demasiadamente do baricentro em certa dimensão, ele não consegue mais afastar-se do mesmo.



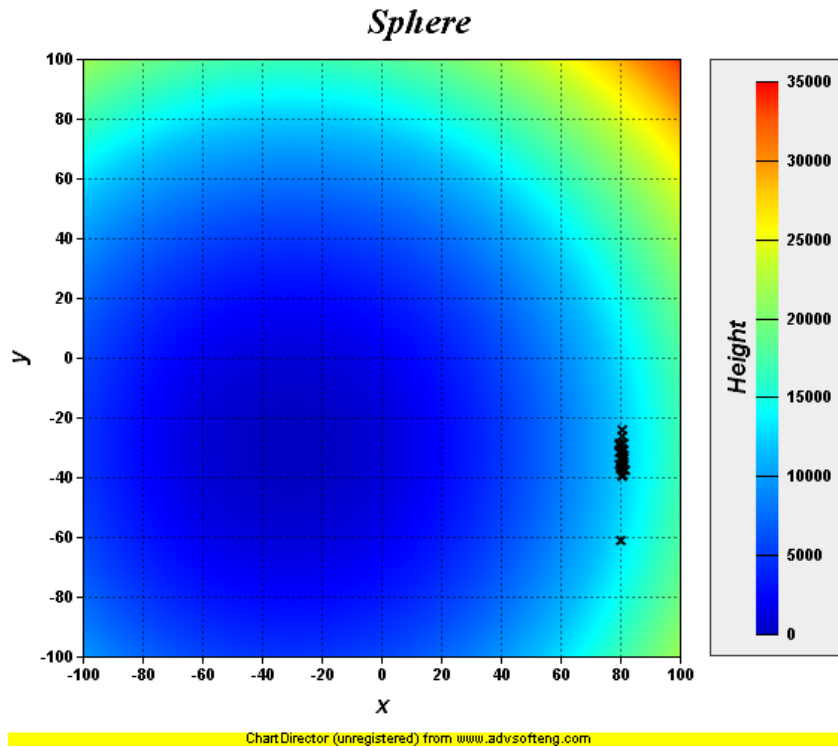


Figura 3.2: Perda de diversidade do cardume na dimensão x. Imagem obtida utilizando-se o projeto Doss criado pelo Grupo de Pesquisa em Inteligência de Enxames - POLI.

## 3.2 Contribuição: FSS II Modificado

Devido aos problemas com o vetor volitivo enfrentado pela versão original do FSS II, decidiu-se criar uma versão modificada para mitigar o problema. Para tanto, reutilizou-se o passo volitivo presente no FSS. Com a presença do passo é possível ter controle da magnitude do vetor volitivo, impedindo assim, a aproximação excessiva dos peixes ao baricentro, o que causa problemas como o mostrado na Figura (3.2). Após realizar essa alteração, notou-se melhora significativa no melhor *fitness* encontrado pelo algoritmo, porém ainda longe da convergência. Por isso, adicionalmente à introdução do passo, várias mudanças foram testadas até que descobriu-se que utilizando a variação do peso do peixe ao invés do seu próprio peso no componente instintivo dado por (2.15), obtinha-se a convergência do algoritmo. Por fim, chegou-se a uma nova Equação para atualização da posição dos peixes dada por:

$$\begin{aligned} \vec{x}_i(t+1) = & \vec{x}_i(t) + \beta \cdot \Delta \vec{x}_i(t+1) + \vec{U}(0,1) \cdot \frac{\sum_{i=1}^n (\Delta \vec{x}_i(t+1) \Delta W_i(t+1))}{\sum_{i=1}^n |\Delta W_i(t+1)|} + \\ & s_{vol} \cdot \vec{U}(0,1) \cdot \text{sign} \left[ \sum_{i=1}^n \Delta W_i(t+1) \right] \cdot \frac{[\vec{x}_i(t) - \vec{B}(t+1)]}{\text{dist}[x_i(t), B(t+1)]}, \end{aligned} \quad (3.1)$$

em que  $\vec{x}_i(t+1)$  é a nova posição do peixe;  $\vec{x}_i(t)$  é sua posição anterior;  $\beta$  é um parâmetro de inércia;  $\Delta\vec{x}_i(t+1)$  é a variação na posição do peixe  $i$  nas duas últimas iterações;  $\Delta W_i(t+1)$  é a variação no peso do peixe  $i$  nas duas últimas iterações;  $dist[\cdot]$  é uma função que retorna a distância euclidiana entre o baricentro e a posição atual do peixe;  $sign[\cdot]$  é uma função que retorna o sinal do argumento e  $\vec{B}(t+1)$  é a posição do baricentro.

As Equações para cálculo de  $\Delta W_i(t+1)$ ,  $\Delta\vec{x}_i(t+1)$  e  $\vec{B}(t+1)$  são expressas respectivamente pelas Equações (2.12), (2.14) e (2.17), as quais são as mesmas utilizadas pela versão sem modificações do FSS II. Reforça-se que o módulo no denominador do componente instintivo ( $|\Delta W_i(t+1)|$ ) é essencial, já que as variações de pesos podem ser positivas ou negativas.

É importante lembrar que dos dois passos presentes no FSS, apenas o volitivo foi introduzido, o que reduz a quantidade de parâmetros necessários para a definição do algoritmo.

### 3.2.1 Análise da Influência dos Operadores

Após obter uma versão convergente do FSS II, o FSS II Modificado, realizou-se uma análise da influência dos operadores. Essa análise foi feita utilizando-se os vetores de movimento dos algoritmos. De fato, essa análise foi essencial antes, durante e após a melhoria do FSS II. Por intermédio dos gráficos gerados, pode-se visualizar o módulo de cada vetor, assim como a proporção entre eles. Para tanto, são apresentados os módulos dos vetores de movimento no decorrer de todas as iterações de busca para o FSS II Original, FSS II Modificado, FSS e PSO Global.

As Figuras (3.3), (3.4), (3.5) mostram respectivamente os vetores individual, instintivo e volitivo do FSS II Original. Esses vetores foram exibidos em figuras diferentes para facilitar a visualização, uma vez que os três vetores possuem módulos semelhantes no decorrer das iterações. A semelhança entre os módulos dos vetores ocorre porque no FSS II Original os operadores (vetores de movimento) individual e instintivo possuem grande dependência modular com relação ao operador volitivo, ou seja, é o vetor volitivo que impulsiona os outros dois. É possível notar também que os três vetores possuem módulo pequeno (entre 0,001 a 0,5) quando comparado com o restante da busca até em torno da iteração três mil, fato que evidencia que a estagnação do cardume na mesma posição durante esse momento da busca. Já a partir da ite-

ração três mil aproximadamente, os módulos dos vetores oscilam entre  $10^{-7}$  a 50 o que não permite a convergência do algoritmo já que não há decaimento com o passar do tempo. Esse não-decaimento, assim como a oscilação no decorrer de todas as iterações deve-se a um comportamento anômalo do parâmetro  $c$ , o qual controla a granularidade da busca. Sem o decaimento, o algoritmo não transita de busca em largura para busca em profundidade com o passar das iterações, dificultando a convergência. De fato, nota-se a convergência nos vetores do FSS, FSS Modificado e PSO Global mostrados pelas Figuras (3.6), (3.7) e (3.8), respectivamente.

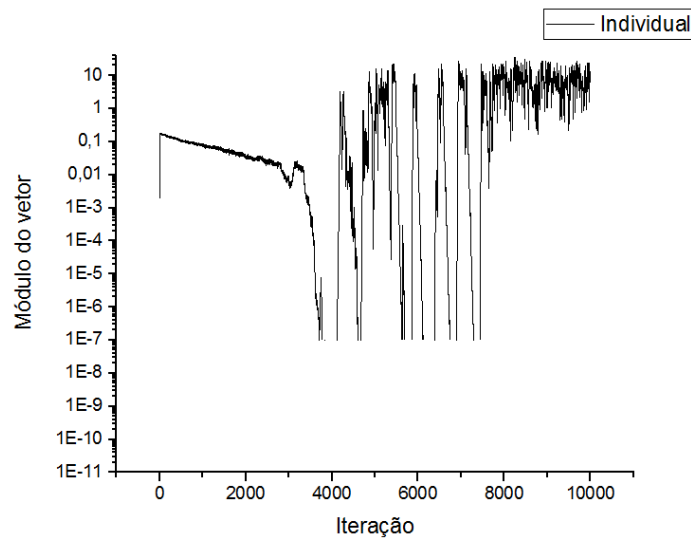


Figura 3.3: Vetor de movimento individual do FSS II Original ao longo das iterações de busca na função Esfera.

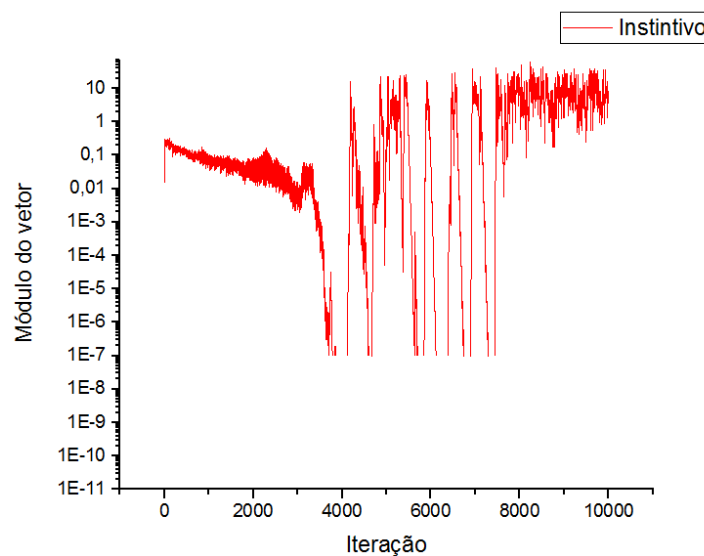


Figura 3.4: Vetor de movimento instintivo do FSS II Original ao longo das iterações de busca na função Esfera.

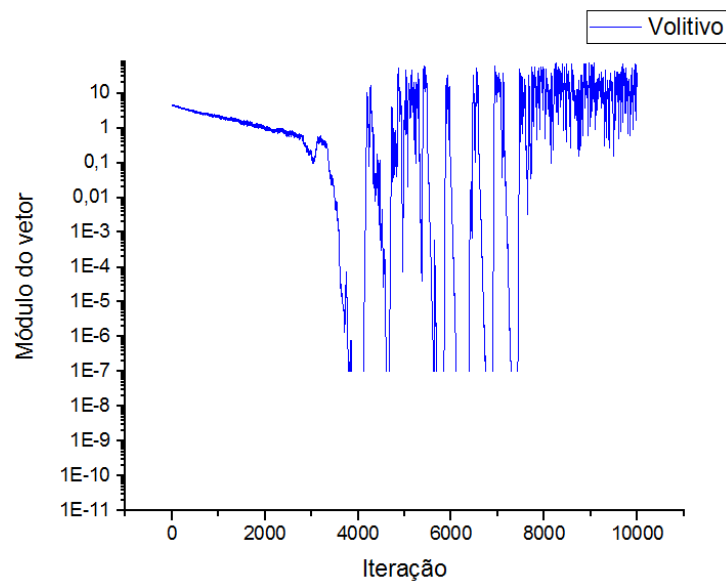


Figura 3.5: Vetor de movimento volitivo do FSS II Original ao longo das iterações de busca na função Esfera.

A trepidação nos vetores do FSS (Figura (3.7)) e do FSS II Modificado (Figura (3.6)) se deve à adição da aleatoriedade por intermédio do vetor  $\vec{U}$ , o qual representa uma distribuição uniforme de probabilidade em um intervalo pré-definido. O único termo sem aleatoriedade é o volitivo do FSS, o que se deve apenas a uma decisão de implementação. Também é possível notar que tanto no FSS como no FSS II Modificado, há um decaimento linear dos vetores. Isso se deve ao uso do passo volitivo, o qual decai linearmente, em ambos os algoritmos. Além disso, tanto no FSS quanto no FSS II Modificado, os módulos dos vetores individuais e instintivos são menores que o módulo do vetor volitivo no decorrer de toda a busca. No FSS, isso ocorre porque para a função Esfera, o passo individual, que determina o módulo dos vetores individual e instintivo, é menor que o passo volitivo no decorrer de todas as iterações. Já no FSS II o mesmo ocorre porque o  $\Delta \vec{x}_i(t)$ , que determina o módulo dos termos individual e instintivo, é sempre menor que o módulo do vetor volitivo. Para ambos algoritmos convergirem, não se faz obrigatório que os vetores individuais e instintivos sejam menores que o volitivo, entretanto, essa configuração foi a que obteve melhores resultados.

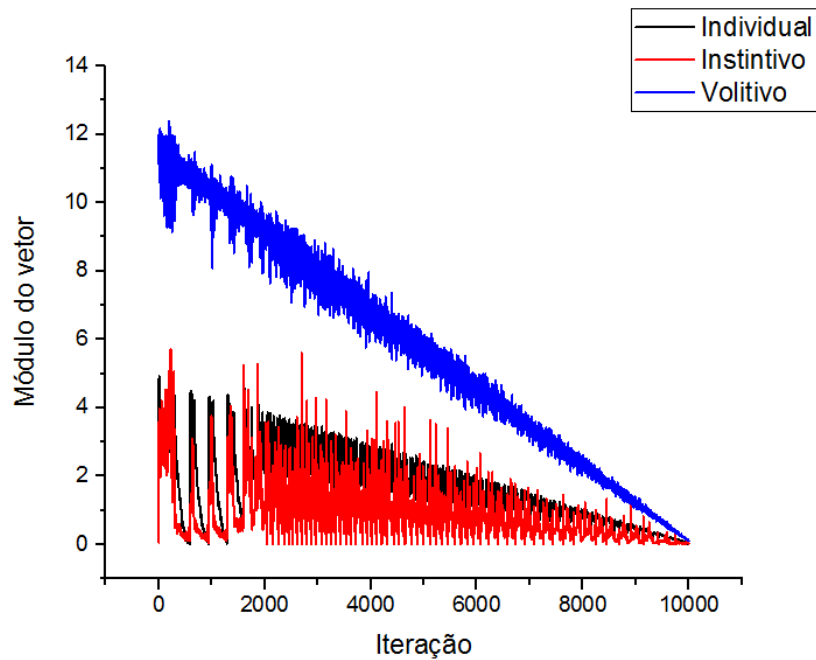


Figura 3.6: Vetores de movimento do FSS II Modificado ao longo das iterações de busca na função Esfera.

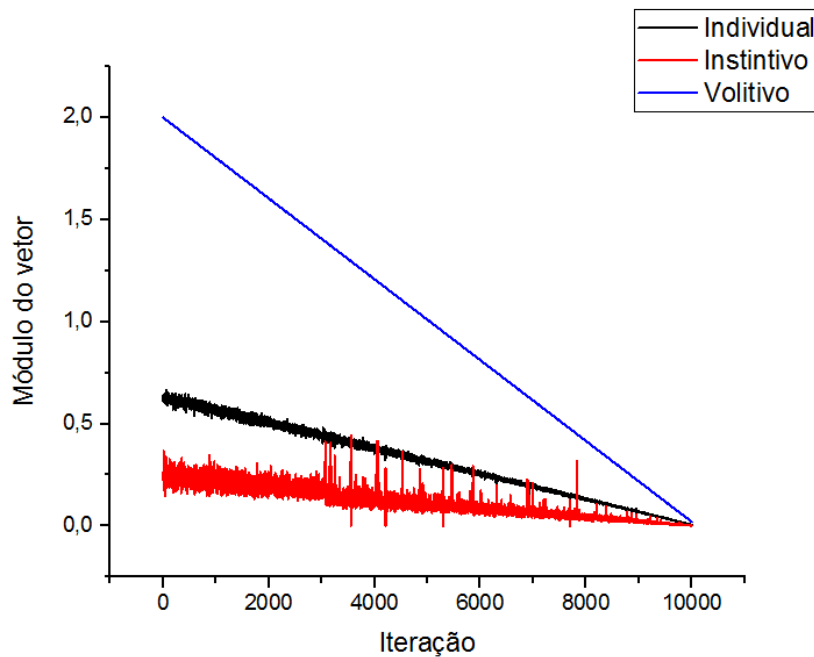


Figura 3.7: Vetores de movimento do FSS ao longo das iterações de busca na função Esfera.

Na Figura 3.8 percebe-se que os módulos dos vetores do PSO decrescem com o passar das iterações, como esperado. Além disso, a configuração paramétrica adotada enfatiza a busca em profundidade, o que pode ser notado por meio dos valores baixos dos módulos. Em contrapartida, ocorre uma convergência prematura do PSO próximo à iteração dois mil, o que pode fazer com que as partículas fiquem presas em

mínimos locais.

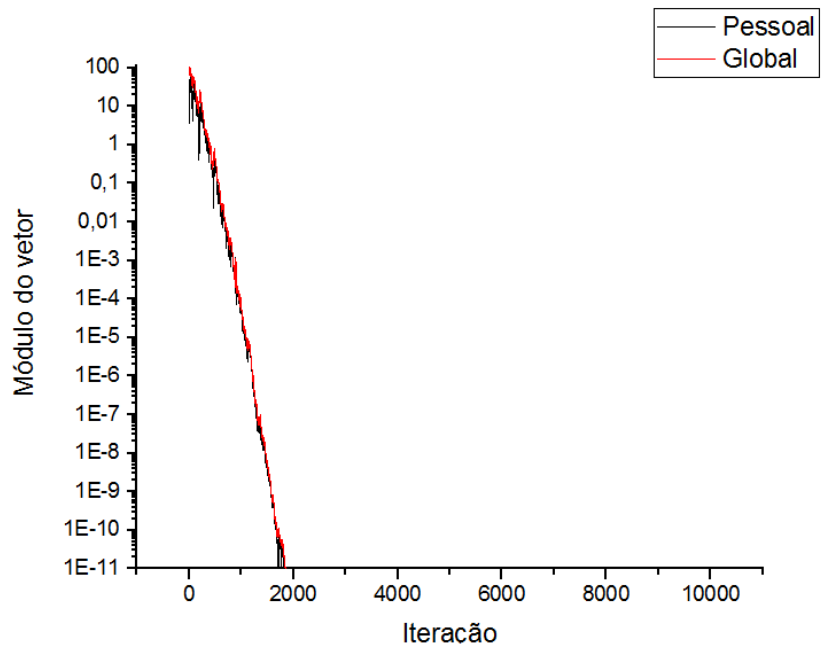


Figura 3.8: Vetores de movimento do PSO-Global ao longo das iterações de busca na função Esfera

Informações importantes foram extraídas da plotagem dos vetores de movimento dos algoritmos. Por isso, essa análise foi de fundamental importância no desenvolvimento desse trabalho.

## 4 *Resultados e Discussão*

*“Nunca será um bom matemático aquele que não for um pouco filósofo.”*

– **Albert Einstein**

Este capítulo tem como objetivo apresentar o arranjo experimental, as funções de teste utilizadas, a configuração paramétrica adotada para cada algoritmo e os resultados das simulações. Além disso, comentários são tecidos a respeito dos resultados obtidos, mostrando que as modificações realizadas no FSS II permitem a sua convergência e competição com o FSS e PSO-G.

### 4.1 Arranjo Experimental

Os algoritmos utilizados para testes comparativos foram FSS II Original [1], FSS II Modificado, FSS [10] e PSO-Global [4]. Foram utilizadas populações de trinta indivíduos, uma busca em trinta dimensões e dez mil iterações como critério de parada. Além disso, as funções de teste Esfera, Rastrigin, Rosenbrock, Griewank e Ackley foram escolhidas e estão representadas pelas Equações (4.1), (4.2), (4.3), (4.4) e (4.5), respectivamente:

$$F_{EsferaDeslocada}(\vec{x}) = \sum_{i=1}^n (x_i + 30)^2, \quad (4.1)$$

$$F_{Rastrigin}(\vec{x}) = 10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)), \quad (4.2)$$

$$F_{Rosenbrock}(\vec{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \quad (4.3)$$

$$F_{Griewank}(\vec{x}) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad (4.4)$$

$$F_{Ackley}(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, \quad (4.5)$$

em que  $i$  representa cada dimensão.

Observa-se que uma variação da Função Esfera a qual possui ponto ótimo  $-30,0^D$  e foi utilizada para também testar os algoritmos com pontos ótimos distantes de  $0,0^D$ . As funções utilizadas possuem características que podem dificultar o processo de convergência dos algoritmos. Por exemplo, a função Esfera é usada para explorar a capacidade de busca inicialmente em largura e depois em profundidade do algoritmo. A função Rastrigin contém vários mínimos locais (multimodal) dos quais os algoritmos precisam escapar. A função Rosenbrock possui um *plateau* em que não é fácil identificar variações na função de *fitness* e assim por diante. As Figuras (4.1), (4.2), (4.3) e (4.4) representam em duas dimensões, algumas das funções utilizadas. Nessas Figuras,  $x$  e  $y$  correspondem às variáveis de entrada e a cor representa o *fitness* do ponto  $(x,y)$ . Quanto maior o valor do ponto, pior é seu *fitness* e vice-versa. As imagens foram obtidas por meio do *framework* Doss, o qual foi utilizado na condução desse trabalho e criado pelo Grupo de Pesquisa em Inteligência de Enxames da Escola Politécnica de Pernambuco.

É importante frisar que a função Esfera Deslocada é uma variação do apresentado na literatura, porém conserva a mesma estrutura da função Esfera original e foi criada pelo autor desse trabalho com o objetivo de obter um ponto ótimo distante da posição  $0,0^D$ .

A Tabela (4.1) mostra a configuração das funções de teste. Percebe-se que todos os peixes ou partículas foram inicializados longe do ponto de mínimo em todos os casos.

## 4.2 Parâmetros utilizados

Para todos os algoritmos, exceto o FSS II Modificado, os parâmetros utilizados foram retirados de artigos.



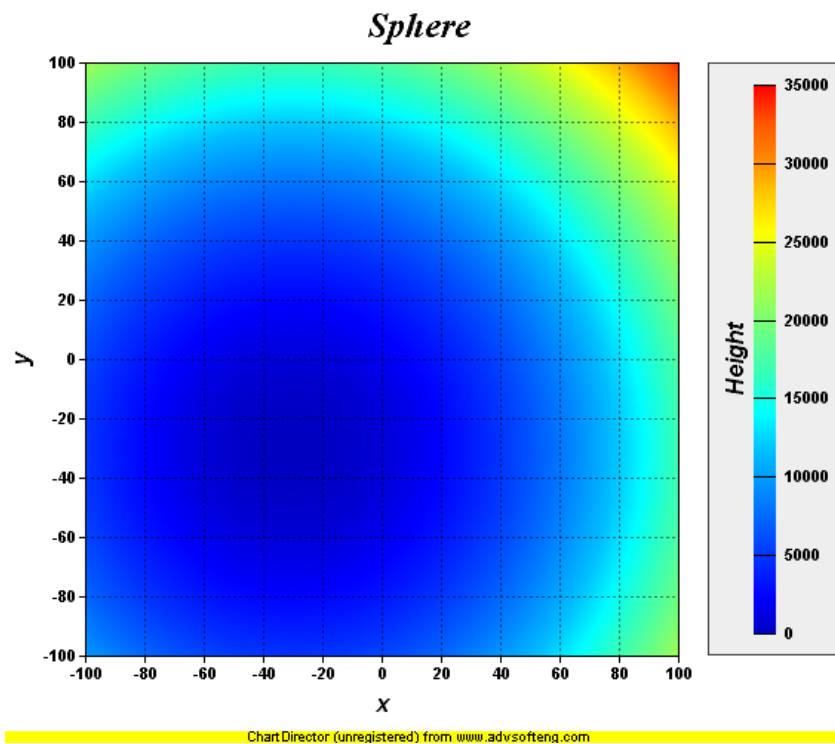


Figura 4.1: Representação Gráfica da Função Esfera em duas dimensões.

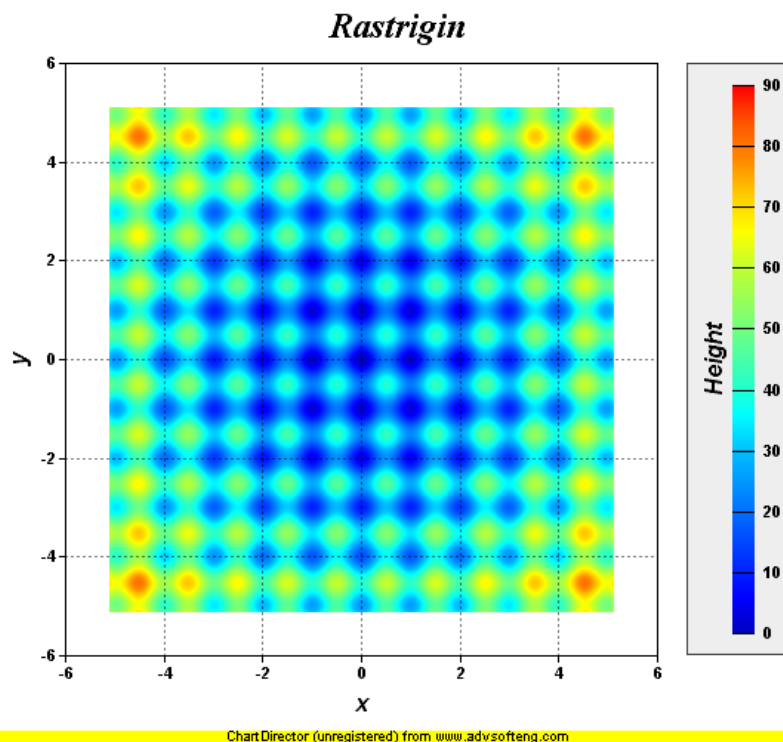


Figura 4.2: Representação Gráfica da Função Rastrigin em duas dimensões.

### 4.2.1 FSS II Original

Assim como em [10], os parâmetros adotados para o Algoritmo de Busca por Car-dumes Versão Dois foram:  $\alpha = 0,01$ ,  $\beta = 0,4$ ,  $c_{inicial} = 0,1$  e  $W_{inicial} = 500$ . Os valores

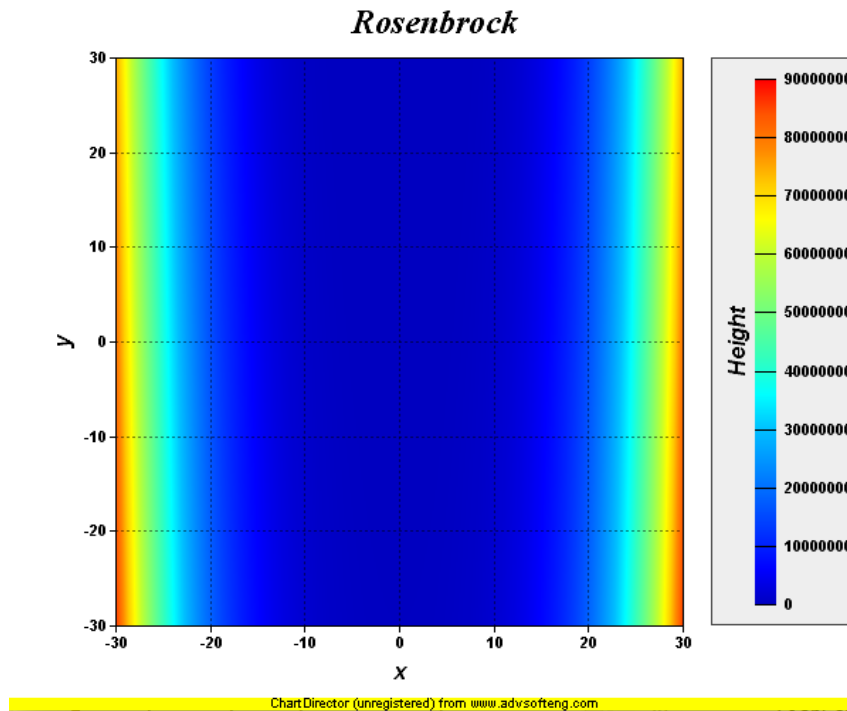


Figura 4.3: Representação Gráfica da Função Rosenbrock em duas dimensões.

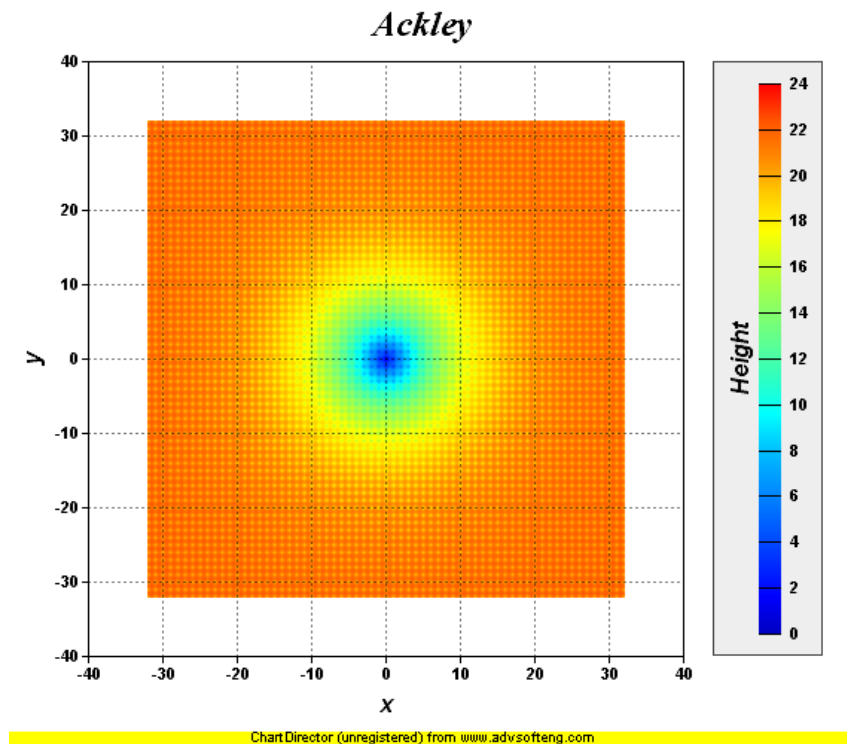


Figura 4.4: Representação Gráfica da Função Ackley em duas dimensões.

de  $c$  tem limite mínimo de  $c_{min} = 0,001$  e máximo de  $c_{max} = 0,999$ , enquanto que o peso varia de  $W_{min} = 1,0$  a  $W_{max} = 1000,0$ .

Tabela 4.1: Espaço de busca, de inicialização e ponto ótimo das funções de teste utilizadas nas simulações.

Função	Espaço de busca	Espaço de inicialização	Ponto ótimo
<b>Esfera</b>	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$	$-30,0^D$
<b>Rastrigin</b>	$-5,12 \leq x_i \leq 5,12$	$2,56 \leq x_i \leq 5,12$	$0,0^D$
<b>Rosenbrock</b>	$-30 \leq x_i \leq 30$	$15 \leq x_i \leq 30$	$1,0^D$
<b>Griewank</b>	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$	$0,0^D$
<b>Ackley</b>	$-32 \leq x_i \leq 32$	$16 \leq x_i \leq 32$	$0,0^D$

### 4.2.2 FSS II Modificado

Para a versão modificada, utilizou-se  $\beta = 0,3$ ,  $s_{vol\_init} = 10\%$  e  $s_{vol\_final} = 0,1\%$  para todas as funções. Os parâmetros  $s_{vol\_init}$  e  $s_{vol\_final}$  são respectivamente os passos volitivos inicial e final. A percentagem do passo volitivo refere-se ao espaço de busca em cada dimensão.

### 4.2.3 FSS

Assim como em [1], os passos são específicos para cada função e podem ser visualizados na Tabela (4.2).

Tabela 4.2: Configuração dos passos volitivo ( $s_{vol}$ ) e individual ( $s_{ind}$ ) para simulação do FSS.

	$s_{vol\_init}$	$s_{vol\_final}$	$s_{ind\_init}$	$s_{ind\_final}$
<b>Esfera</b>	1%	0,01%	0,1%	0,001%
<b>Rastrigin</b>	10%	0,1%	10%	0,01%
<b>Rosenbrock</b>	1%	0,01%	0,1%	0,001%
<b>Griewank</b>	1%	0,01%	1%	0,001%
<b>Ackley</b>	10%	0,1%	10%	0,01%

### 4.2.4 PSO-G

A configuração do PSO Global deu-se conforme recomendado em [12], isto é, com  $w = 0,729844$ ,  $c_1 = 1,49445$  e  $c_2 = 1,49445$ . Onde  $w$  é o valor da inércia,  $c_1$  é o fator de aprendizagem do componente cognitivo e  $c_2$  é o fator de aprendizagem do componente social.

### 4.3 Simulações

Foram realizadas 30 simulações para cada par: função-algoritmo. Os resultados em termos de melhor *fitness* obtido e desvio padrão, podem ser visualizados na Tabela (4.3). Observa-se que em todas as funções testadas, o desempenho do FSS II Modificado foi superior ao do FSS II Original. Além disso, o PSO-G obteve o melhor *fitness* para a função Esfera, o FSS II Modificado alcançou o melhor *fitness* para as funções Rastrigin e Rosenbrock e por fim, o FSS obteve o melhor *fitness* para as funções Griewank e Ackley.

Tabela 4.3: Valores de melhor *fitness* e desvio padrão correspondente. O desvio padrão de cada função está entre parêntesis.

	<b>FSS 2 Original</b>	<b>FSS 2 Modificado</b>	<b>FSS1</b>	<b>PSO-G</b>
<b>Esfera</b>	62796,677 (15810,97)	1,616 (1,53)	0,004 (0,003)	<b>0</b> <b>(0)</b>
<b>Rastrigin</b>	258,231 (23,85)	<b>21,786</b> <b>(5,72)</b>	53,332 (12,22)	255,802 (50,45)
<b>Rosenbrock</b>	$2,06 \times 10^8$ ( $6,73 \times 10^7$ )	<b>40,667</b> <b>(39,93)</b>	53,532 (10,64)	215,394 (763,81)
<b>Griewank</b>	483,398 (195,44)	0,770 (0,24)	<b>0,025</b> <b>(0,03)</b>	0,0550751 (0,07)
<b>Ackley</b>	19,595 (0,55)	2,463 (0,61)	<b>0,183</b> <b>(0,09)</b>	19,795 (0,04)

Também foram gerados gráficos do *fitness* em função do número de iterações com o objetivo de visualizar a evolução do *fitness* ao longo das iterações. Nesses gráficos, o valor do *fitness* em cada iteração é a média aritmética do valor do *fitness* de mesma iteração em trinta simulações. Em algumas funções, utilizou-se escala logarítmica para permitir melhor visualização. Os resultados para as funções Esfera, Rastrigin, Rosenbrock, Griewank e Ackley podem ser visualizados nas Figuras (4.5), (4.6), (4.7), (4.8) e (4.9), respectivamente.

Na Figura (4.5), que mostra a evolução do valor do melhor *fitness* para a função Esfera, nota-se que o PSO-G obteve o melhor valor de *fitness*. É importante salientar que a configuração adotada contribui para o bom resultado, uma vez que ela permite ao PSO buscar mais em profundidade que em largura. Sendo assim, nessa função, o PSO move-se para o mínimo global e inicia uma forte busca em profundidade. Já o FSS e o FSS II Modificado, os quais são controlados pelo passo, realizam uma busca em profundidade limitada. Por isso, esses dois algoritmos obtém o segundo e terceiro melhor *fitness* respectivamente. Por fim, o FSS II Original permanece praticamente

com o mesmo valor de melhor *fitness* durante toda a busca. Um detalhe importante para esclarecer os resultados é que o PSO-G obteve valores de *fitness* tão próximos a zero que não foi possível plotar a continuação de sua curva de *fitness* devido à escala logarítmica.

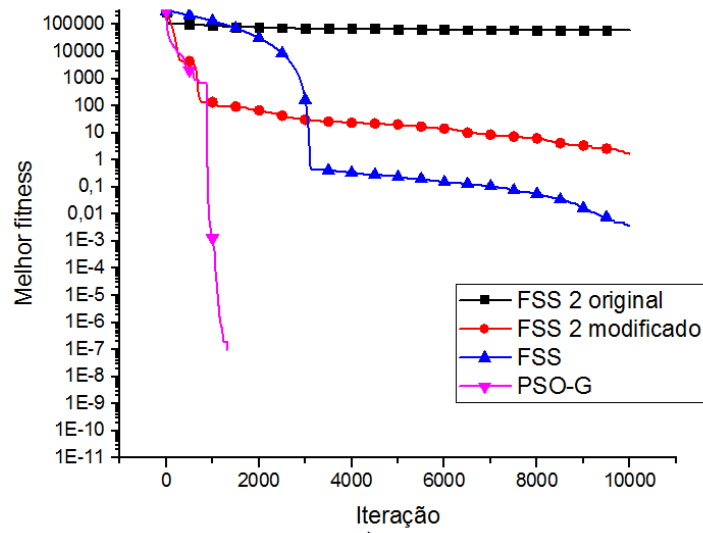


Figura 4.5: Evolução do valor de melhor *fitness* para a função Esfera em dez mil iterações e trinta dimensões.

Na função Rastrigin (Figura (4.6)), o FSS II Modificado obtém o melhor *fitness* seguido pelo FSS. Já o PSO-G converge para um dos mínimos locais obtendo *fitness* semelhante ao do FSS II Original. Essa é a única função onde se nota o decaimento da curva de melhor *fitness* do FSS II Original, devido apenas à escala no eixo y.

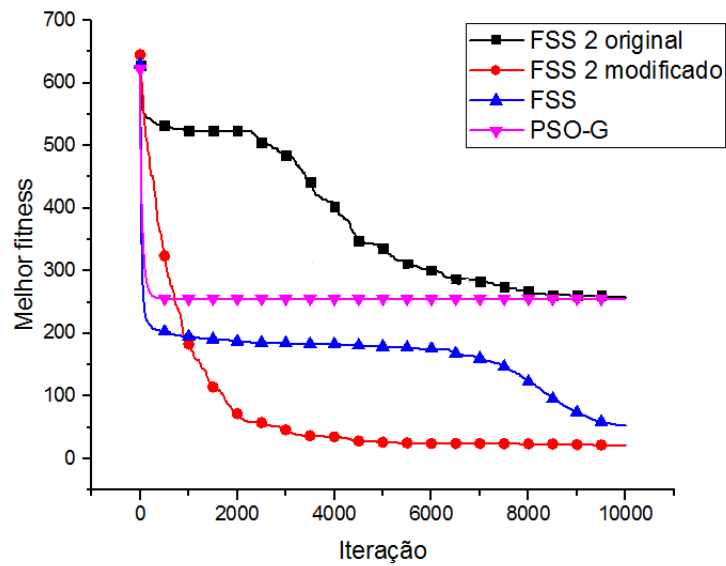


Figura 4.6: Evolução do valor de melhor *fitness* para a função Rastrigin em trinta dimensões e dez mil iterações.

Na função Rosenbrock (Figura (4.7)), é possível notar que os algoritmos FSS, FSS II Modificado e PSO-G possuem valores de *fitness* parecidos. Isso provavelmente se deve ao fato de que esses algoritmos foram atraídos à região de vale da função, mas permaneceram praticamente estagnados após alcançar o vale, o que é exatamente o desafio de otimização da função Rosenbrock.

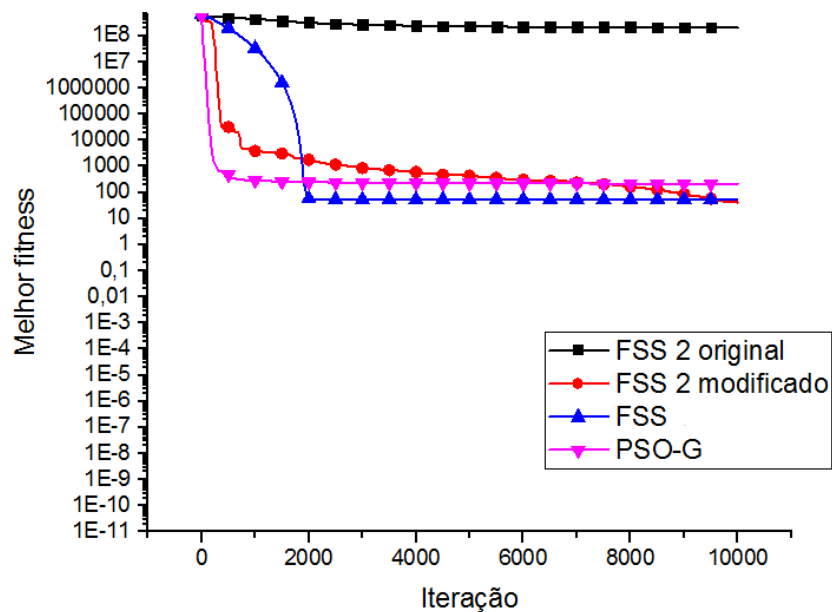


Figura 4.7: Evolução do valor de melhor *fitness* para a função Rosenbrock em trinta dimensões e dez mil iterações.

A evolução do valor de melhor *fitness* da função Griewank, mostrada na Figura

(4.8), revela que o melhor valor de *fitness* é obtido pelo FSS, que supera o *fitness* do PSO-G no momento final da busca ao passo que o FSS II Modificado obtém o terceiro melhor *fitness*. O FSS II Original obtém o pior *fitness* dos quatro.

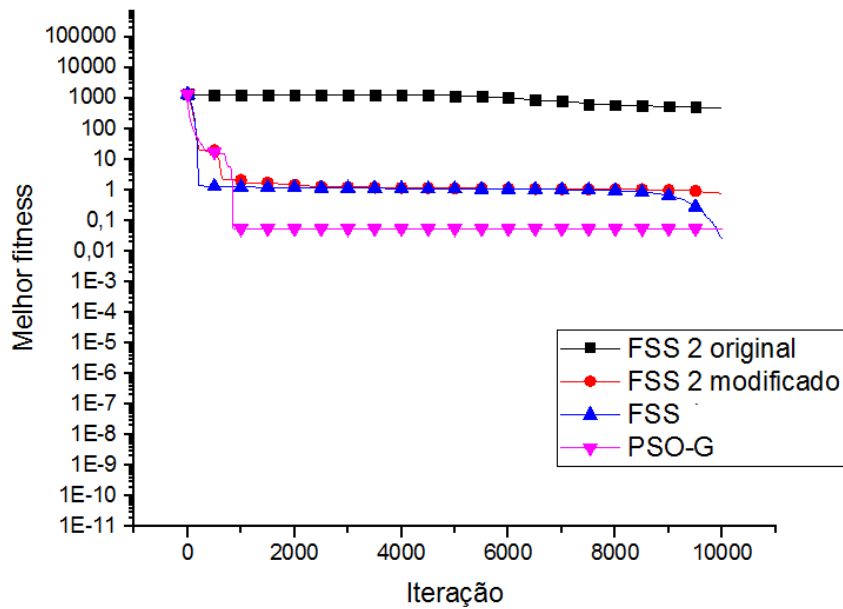


Figura 4.8: Evolução do valor de melhor *fitness* para a função Griewank em trinta dimensões e dez mil iterações.

Na função Ackley, o FSS supera o FSS II Modificado no momento final da busca enquanto que o FSS II Original e o PSO-G obtém valores de *fitness* semelhantes.

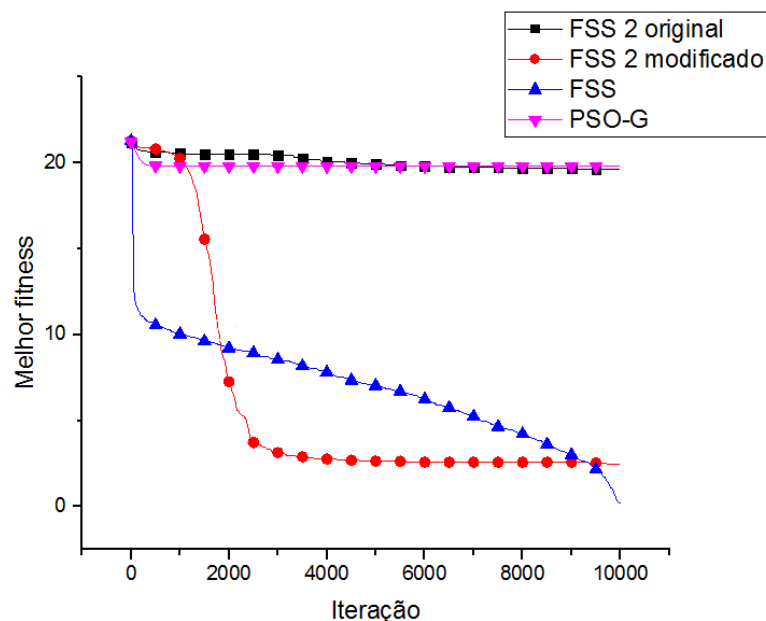


Figura 4.9: Evolução do valor de melhor *fitness* para a função Ackley em trinta dimensões e dez mil iterações.

Os gráficos *boxplot* foram gerados com o resultado de melhor *fitness* obtidos em trinta simulações. Em alguns casos, utilizou-se novamente escala logarítmica no eixo y para melhorar a visualização.

No *boxplot* da função Esfera, mostrado na Figura (4.10), todos os valores de *fitness* do PSO-G são iguais a 0. De fato, os resultados obtidos pelo PSO-G para a função Esfera foram excelentes e devem-se também ao fato que a função Esfera é uma função com apenas um ponto de mínimo, o que favorece à configuração paramétrica e à topologia Global aplicadas no PSO. O FSS obtem o segundo melhor *fitness* com uma pequena dispersão, seguido pelo FSS II Modificado também com uma pequena dispersão e por fim pelo FSS II Original, o qual possui resultados em torno de  $10^5$ .

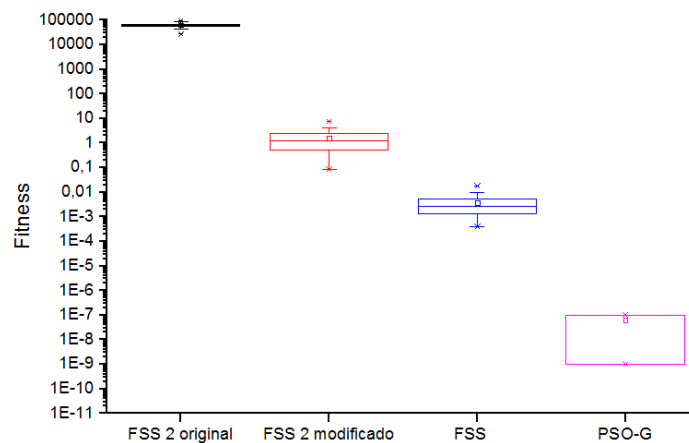


Figura 4.10: Boxplot dos resultados de *fitness* para a Função Esfera em trinta dimensões e dez mil iterações.

No *boxplot* da função Rastrigin (Figura (4.11)), é possível observar que o FSS II Modificado obtém os melhores resultados, sendo seguido pelo FSS, PSO-G e FSS II Original, respectivamente. O PSO-G possui valor de mediana próximo ao do FSS II Original, mas com uma dispersão maior que o último.



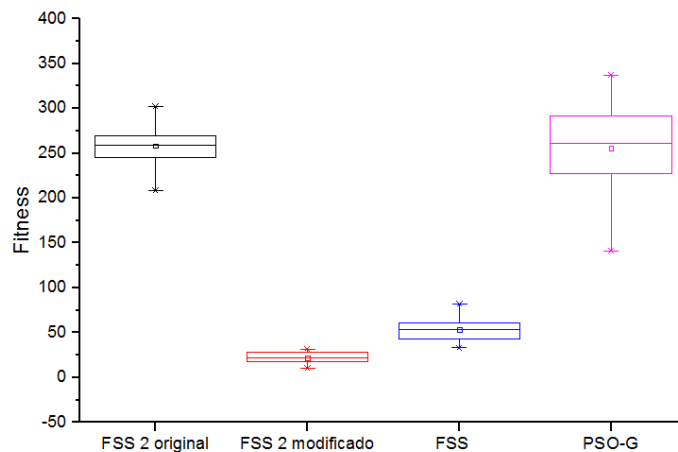


Figura 4.11: Boxplot dos resultados de *fitness* para a Função Rastrigin em trinta dimensões e dez mil iterações.

No *boxplot* da função Rosenbrock (Figura (4.12)) nota-se que o FSS II Modificado e o FSS obtêm valores muito próximos, porém a maioria dos valores do FSS II Modificado são ligeiramente menores que os do FSS. Já o PSO-G obtém o melhor valor individual de *fitness*, porém possui maior média, já que a dispersão é alta, possuindo o terceiro melhor *fitness*. Por fim, o pior resultado é obtido pelo FSS II Original, que está bem longe dos outros.

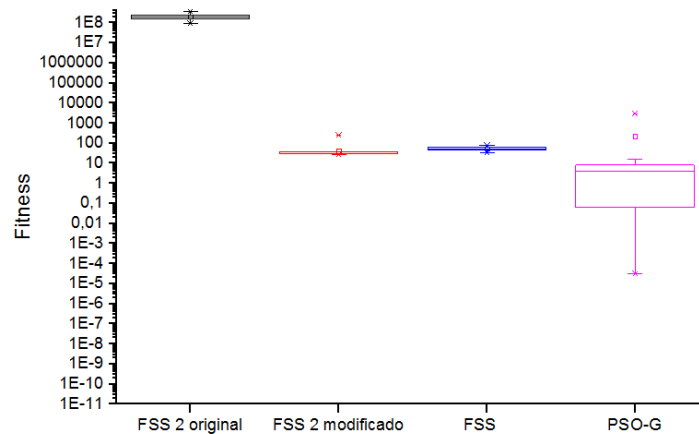


Figura 4.12: Boxplot dos resultados de *fitness* para a Função Rosenbrock em em trinta dimensões e dez mil iterações.

No *boxplot* da função Griewank (Figura (4.13)), o PSO-G obtém o melhor valor individual de *fitness*, entretanto possui média maior que a do FSS em virtude do alto desvio padrão. O FSS II Modificado segue na terceira posição acompanhado pelo FSS II Original, que possui os piores resultados.

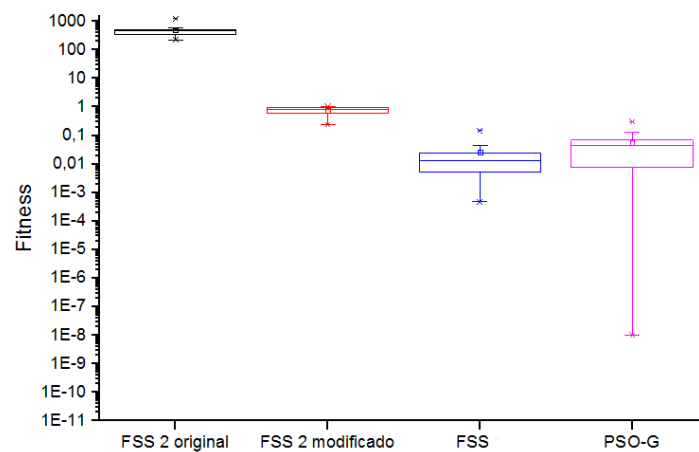


Figura 4.13: Boxplot dos resultados de *fitness* para a Função Griewank em trinta dimensões e dez mil iterações.

No *boxplot* da função Ackley, o FSS e o FSS II Modificado obtém valores próximos e o FSS alcança os melhores valores. Já o FSS II Original e o PSO-G obtém resultados semelhantes.

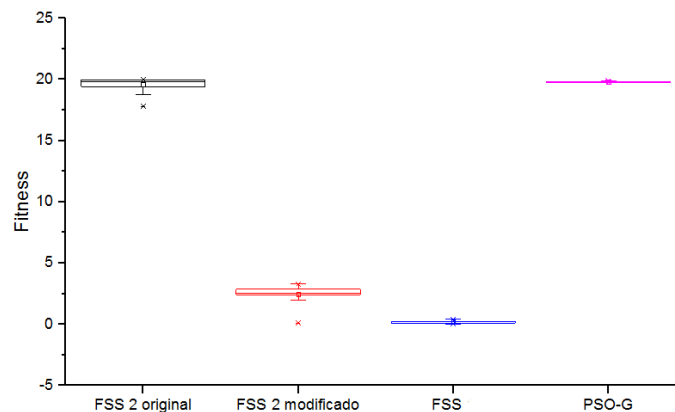


Figura 4.14: Boxplot dos resultados de *fitness* para a Função Ackley em trinta dimensões e dez mil iterações.

É possível notar em todas as funções, a rápida convergência do PSO-G devido à sua topologia e configuração de parâmetros. O FSS II Original, em todas as funções, exceto Rastrigin, obtém resultados não-satisfatórios. Já o FSS e o FSS II Modificado, obtém resultados bons e semelhantes em todas as funções.

Por meio dos gráficos *boxplot*, constata-se que o desvio padrão é pequeno tanto no FSS como no FSS II Modificado, o que mostra que o resultado de ambos são consistentes. Nota-se que o FSS II Modificado superou o FSS II Original em todas as funções de teste e o FSS apenas na Função Rastrigin (Figura (4.11)), empatando na Função Rosenbrock (Figura (4.12)) e perdendo na Função Esfera (Figura (4.10)),

Griewank (Figura (4.13))e Ackley (Figura (4.14)). Apesar de perder na maioria dos casos para o FSS, o FSS II Modificado manteve-se competitivo com o mesmo.

## 5 *Conclusões e Trabalhos Futuros*

*”Os computadores são incrivelmente rápidos, precisos e burros;  
os homens são incrivelmente lentos, imprecisos e brilhantes;  
juntos, seus poderes ultrapassam os limites da imaginação.”*

**– Albert Einstein**

Neste capítulo, serão feitas considerações finais sobre os resultados obtidos e, por fim, serão apresentados os trabalhos futuros.

### 5.1 **Conclusões**

Algoritmos de otimização garantem soluções aceitáveis para problemas complexos com alta dimensionalidade, combinações não lineares e espaços de busca amplos. O FSS II é um algoritmo de otimização derivado do FSS e criado com o propósito de retirar limitações presentes no FSS como o uso de passos e o alto custo computacional devido à necessidade de avaliar o *fitness* duas vezes por peixe por iteração. Entretanto, o FSS II possui problemas de convergência em certos casos. Os problemas devem-se à necessidade de ajustes de parâmetros e operadores do FSS II.

A contribuição desse trabalho deu-se de duas formas. A primeira foi a identificação dos parâmetros e operadores responsáveis pelos problemas de convergência do FSS II. O parâmetro que apresenta comportamento não esperado é o  $c$  e o operador é o volitivo. Notou-se que devido ao  $c$  os módulos dos vetores não decrescem com o decorrer da busca e que devido à configuração do termo volitivo no FSS II, há uma convergência prematura no baricentro em pelo menos uma das dimensões. A segunda contribuição ocorreu na forma de modificações que mitigam o problema. Essas mudanças consistem no retorno do passo volitivo e no uso da variação do peso do peixe ao invés do valor absoluto do peixe no termo instintivo. Assim, o FSS II Modificado foi

criado.

Para testar a eficácia da versão modificada do FSS II foram selecionadas funções de teste amplamente utilizadas na comunidade científica como Esfera, Rastrigin, Rosenbrock, Griewank e Ackley. Inclusive, modificou-se a função Esfera para que a mesma tivesse o ponto de mínimo distante do zero com o intuito de garantir maior confiabilidade aos testes já que todas as outras funções de teste utilizadas possuem ponto de mínimo no zero ou em regiões próximas ao zero. Além disso, a experimentação foi realizada de forma que garante a validade estatística da mesma. Assim, foi possível mostrar que o FSS II Modificado obteve melhor desempenho nas funções de teste utilizadas que o FSS II Original. Além disso, o FSS II Modificado é computacionalmente mais eficiente que o FSS uma vez que precisa de apenas uma avaliação de *fitness* por peixe por iteração (o FSS precisa de duas) e só usa um parâmetro estático ( $\beta$ ) e um passo (o volitivo) enquanto que o FSS precisa de dois passos.

## 5.2 **Trabalhos Futuros**

O FSS é uma família de algoritmos de otimização por enxames que merece estudos mais aprofundados devido ao seu sucesso e ao fato de ser uma família de algoritmos ainda recém-nascida.

Uma melhoria a ser investigada é o ganho normalizado de peso. A normalização impede que o algoritmo diferencie uma região muito boa de outra muito ruim, pois independente de onde os peixes estiverem, o ganho no peso sempre será de -1 a 1. Conseqüentemente, peixes que se distanciam de uma região ruim e aproximam-se de uma boa podem não ser capazes de guiar o algoritmo para a região melhor. Uma possível solução seria retirar a normalização e permitir que os pesos mínimo e máximo dos peixes possam variar em um intervalo maior.

Também é desejável uma comparação futura do FSS e do FSS II Modificado com outras configurações e topologias do PSO assim como com outros algoritmos, como por exemplo o ABC. Esses algoritmos e configurações adicionais permitirão uma comparação mais robusta entre as diferentes versões do FSS e outros algoritmos de otimização de enxames. Funções de teste mais recentes também podem ser incorporadas para testar cenários diferentes.

Além disso, busca-se ainda por uma versão do FSS II sem passos, o que permi-

tirá uma maior facilidade de implementação e uma maior independência do algoritmo. Uma possibilidade de implementação sem passos é a hibridização do FSS com o PSO onde os peixes usarão não só o baricentro, mas também o *personal best* como referência no processo de busca. Acredita-se que essa abordagem impedirá a convergência prematura no baricentro. Outras inspirações podem ser buscadas em outras famílias de algoritmos de enxames existentes ou até no próprio cardume real.

## *Referências Bibliográficas*

- [1] BASTOS-FILHO, C. J. A. et al. A novel search algorithm based on fish school behavior. *IEEE International Conference on Systems, Man, and Cybernetics*, p. 2646–2651, 2008.
- [2] ENGELBRECHT, A. *Computational Intelligence An Introduction*. [S.l.]: Wiley & Sons, 2007.
- [3] BONEBEAU, E.; DORIGO, M.; THERAULAZ, G. *Computational Intelligence An Introduction*. [S.l.]: Oxford University Press, 1999.
- [4] KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. Perth, Aust: IEEE, 1995. p. 1945–1948.
- [5] DORIGO M., M. V.; COLORNI, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst., Man, and Cybern. – Part B*, 26(1), 29, 1996.
- [6] C., E. R.; SHI, Y. *Computational Intelligence: Concepts to implementations*. 1 ed. ed. Burlington, MA, EUA: Morgan Kaufmann, 2007.
- [7] PANIGRAHI, B. K.; SHI, Y.; LIM, M. H. *Handbook of Swarm Intelligence: Concepts, principles and applications*. Burlington, MA, EUA: Springer, 2011.
- [8] HELLINGRATH, B. et al. Application of the fish school search for the supply chain network planning problem. *Anais do XI Congresso Brasileiro de Inteligência Computacional*, p. 14–19, 2013.
- [9] YUAN, X. H. et al. A survey on application of particle swarm optimization to electric power systems. *IEEE Int. Congress Evolutionary Computation*, p. 14–19, 2004.
- [10] BASTOS-FILHO, C. J. A.; NASCIMENTO, D. O. An enhanced fish school search algorithm. *1st BRICS Countries Congress (BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence*, 2013.
- [11] LACERDA, M. G. P. *Weight-Based Approach for Niching in Fish School Search*. Dissertação (Mestrado) — POLI, Escola Politécnica de Pernambuco, UPE, 2014.
- [12] CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *Proc. ICEC, Washington, DC*, p. 1951–1957, 1999.