



COPPE/UFRJ

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS
COM CONGREGAÇÃO PASSIVA SELETIVA

Jorge Viot

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientador: Nelson Francisco Favilla Ebecken

Rio de Janeiro
Março de 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS
COM CONGREGAÇÃO PASSIVA SELETIVA

Jorge Viot

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. Alexandre Gonçalves Evsukoff, DR.

Prof. Luiz Landau, D.S.c

Prof. Hélio José Côrrea Barbosa, D.Sc.

Prof^ª. Fernanda Araujo Baião, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2010

Viot, Jorge

Otimização por Enxame de Partículas com Congregação Passiva Seletiva/ Jorge Viot. – Rio de Janeiro: UFRJ/COPPE, 2010.

XI, 94 p.: il.; 29,7 cm.

Orientador: Nelson Francisco Favilla Ebecken

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2010.

Referencias Bibliográficas: p. 79-87

1. Otimização.
2. Computação Evolutiva.
3. Otimização por Enxame de Partículas. I. Ebecken, Nelson Francisco Favilla. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

DEDICATÓRIA

Dedico este trabalho a meus
pais Domingos e Marlene
pela base moral e social
sempre presente em todos
os momentos.

AGRADECIMENTOS

Agradeço:

Aos meus pais pela formação que me proporcionaram e pela ajuda, compreensão e apoio; à minha filha Flávia pelo carinho e paciência; e ao meu irmão Maurício, sempre pronto a ajudar.

Ao meu orientador, Professor Nelson Ebecken, pela acolhida, ensinamentos transmitidos, paciência, estímulo e confiança no trabalho por mim proposto. Pelo seu bom humor e simplicidade de objetivos, sua experiência e inteligência na condução deste trabalho.

À Patricia pela sua chegada em um momento importante da minha vida e deste trabalho.

Ao Exmo. Sr. Contra-Almirante Liseo Zampronio, atual diretor do CASNAV (Centro de Análises de Sistemas Navais) e representando todos os demais, pelo apoio incondicional sem o qual não seria possível a realização do curso.

Aos “Casnavianos”, pelo apoio e incentivo durante a realização do curso, em especial à Sandra Tavares sempre pronta a ajudar nas formatações e ao colega Menezes pela disponibilidade em fazer as correções ortográficas.

Aos companheiros da Divisão de Pesquisa Operacional: Espenchitt, Leonardo Claro, Mauricio Guedes, Miguel, Lincoln, Renato Porthun, Geiza, Trevisan, Comte. Malheiros, Comte. Régula, Comte. Bodini, Fernando Ayres, Ronaldo, Denise, Silvio, Lamarca, Leandro, Marco Aurélio, Viviane e Bruno.

À companheira Fernanda pelo apoio incondicional à reestruturação da Divisão de Planejamento e Controle de Projetos.

Aos funcionários da COPPE/PEC, em especial à Eгна e sua abnegada dedicação, ao Jairo Leite, e à Elisabeth que sempre estavam prontos a atender as solicitações inerentes à vida acadêmica.

E, acima de tudo, a Deus, por mais esta etapa vencida.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.).

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS COM CONGREGAÇÃO PASSIVA SELETIVA

Jorge Viot
Março/2010

Orientador: Nelson Francisco Favilla Ebecken
Programa: Engenharia Civil

A Otimização por Enxame de Partículas é um algoritmo de otimização baseado em população que tem obtido relevado reconhecimento na solução de problemas de grande dimensionalidade, com simplicidade e utilizando pequenos recursos computacionais. Seu método operacional está baseado na cooperação e interação social entre os indivíduos do enxame na obtenção de um objetivo comum.

O objetivo deste trabalho é propor a utilização de um mecanismo de comportamento biológico de grupos, denominado congregação passiva, com um processo seletivo, no modelo de otimização por enxame de partículas (PSO) para aumentar o desempenho na solução de problemas de otimização global, combatendo o problema da convergência prematura pelo equilíbrio das fases de exploração global e local.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PARTICLE SWARM OPTIMIZATION
WITH SELECTIVE PASSIVE CONGREGATION

Jorge Viot
March/2010

Advisor: Nelson Francisco Favilla Ebecken

Department: Civil Engineering

The Particle Swarm Optimization (PSO) is a population-based optimization algorithm which has obtained great recognition in solving multi-dimensional problems with simplicity and using few computational resources. Its operational method is based in the cooperation and social interaction among members from the swarm to reach a common objective.

The aim of this work is to propose the use of a mechanism of biological behaviour in groups, named passive congregation, with a selective process in the PSO model, in order to increase the performance in solving problems of global optimization, addressing the premature convergence problem by using the balance of the local and global exploration phases.

ÍNDICE

1 - INTRODUÇÃO	1
1.1 - MOTIVAÇÃO	1
1.2 - OBJETIVO.....	2
1.3 - ORGANIZAÇÃO DA TESE	2
2 - OTIMIZAÇÃO	4
2.1 - ELEMENTOS DO PROCESSO DE OTIMIZAÇÃO.....	5
2.2 - SOLUÇÕES E COMPLEXIDADE DE ALGORITMOS	6
2.3 - MÉTODOS DE RESOLUÇÃO DE PROBLEMAS.....	7
2.3.1 – MÉTODOS EXATOS.....	7
2.3.2 – MÉTODOS APROXIMATIVOS	9
2.3.3 – MÉTODOS HEURÍSTICOS.....	9
3 - COMPUTAÇÃO EVOLUCIONÁRIA.....	13
3.1 - ALGORITMOS EVOLUTIVOS.....	14
3.1.1 – ALGORITMOS GENÉTICOS	18
3.1.2 – PROGRAMAÇÃO EVOLUCIONÁRIA	19
3.1.3 – ESTRATÉGIAS DE EVOLUÇÃO	20
3.1.4 – PROGRAMAÇÃO GENÉTICA	20
4 - INTELIGÊNCIA DE ENXAMES	21
4.1 - OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS	22
4.1.1 – MODELO ANT SYSTEM.....	23
4.1.2 – MODELO ANT COLONY SYSTEM	23
5 - OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS.....	25
5.1 - ENXAME DE PARTÍCULAS É UM TIPO DE ALGORITMO EVOLUCIONÁRIO?.....	25
5.2 - HISTÓRICO	26
5.3 - O ALGORITMO DA OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	28
5.3.1 – INÉRCIA	33
5.3.2 – VARIAÇÃO LINEAR DA INÉRCIA	35
5.3.3 – VARIAÇÃO NÃO LINEAR DA INÉRCIA.....	35
5.3.4 – VARIAÇÃO FUZZY DA INÉRCIA.....	36
5.3.5 – COEFICIENTE DE CONSTRIÇÃO	37
5.3.6 – OUTRAS VARIAÇÕES NA PSO	38
5.3.7 – OS PARÂMETROS DO ALGORITMO DA PSO	39
5.4 - SOCIOMETRIA DE ENXAME DE PARTÍCULAS	40
5.4.1 – TOPOLOGIAS ESTÁTICAS	41
5.4.2 – TOPOLOGIAS DINÂMICAS	46
5.5 - VARIANTES DA PSO	50
5.5.1 – PSO HÍBRIDO.....	50
5.5.2 – PSO ADAPTATIVO	51
5.5.3 – PSO EM AMBIENTES COMPLEXOS.....	51
5.5.4 – PSO DISCRETO	51
5.5.5 – OUTRAS VARIANTES DA PSO	51
6 – PROPOSTA.....	53
6.1 - CONTEXTO DA PROPOSTA.....	53
6.2 - CONGREGAÇÃO PASSIVA SELETIVA.....	54
6.2.1 - PSEUDO-CÓDIGO DA PSO COM CONGREGAÇÃO PASSIVA SELETIVA.....	58

6.3 - ESQUEMA EXECUÇÃO-GERAÇÃO-PARTÍCULA-DIMENSÃO	59
6.4 – AVALIAÇÃO DO ALGORITMO	61
6.4.1 – PARÂMETROS INTERNOS DOS TIPOS DE PSO	61
6.4.2– PARÂMETROS DE CONTROLE DO TESTE	62
6.5 – FUNÇÕES DE TESTE	63
6.6 – PROGRAMA DO EXPERIMENTO	64
6.7– CALIBRAÇÃO DO PARÂMETRO C3 DA PSO-CPS.....	65
6.8 – RESULTADOS EXPERIMENTAIS	67
6.9 – ANÁLISE DE RESULTADOS	75
6.9.1 – COMPLEMENTAÇÃO DE RESULTADOS	76
7 - CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS.....	78
REFERÊNCIAS BIBLIOGRÁFICAS	79

ÍNDICE DE FIGURAS

Figura 1 - Métodos de resolução de problemas.....	7
Figura 2 - Classificação Geral dos Métodos Clássicos [NEVES,1997].....	8
Figura 3 - Classificação das Metaheurísticas [DRÉO, 2007], modificada pelo autor dessa tese.	12
Figura 4 – Representação gráfica de (a) GBEST e (b) LBEST.....	30
Figura 5 – Estrutura básica de um algoritmo PSO	32
Figura 6 – Gráfico da variação não linear do peso da inércia	36
Figura 7 – Representação gráfica das topologias (a) Von Neumann e (b) Roda.....	43
Figura 8 – Gráfico de valor médio da função Sphere.....	69
Figura 9 - Gráfico de melhor valor da função Sphere	69
Figura 10 – Gráfico de valor médio da função Rosenbrock.....	70
Figura 11 – Gráfico de melhor valor da função Rosenbrock	70
Figura 12 – Gráfico de valor médio da função Griewank	71
Figura 13 – Gráfico de melhor valor da função Griewank.....	71
Figura 14– Gráfico de valor médio da função Rastrigin	72
Figura 15 – Gráfico de melhor valor da função Rastrigin.....	72
Figura 16 – Gráfico de valor médio da função Ackley	73
Figura 17 – Gráfico de melhor valor da função Ackley	73
Figura 18 – Gráfico de valor médio da função Schwefel	74
Figura 19 – Gráfico de melhor valor da função Schwefel.....	74

ÍNDICE DE TABELAS

Tabela 1 – Esquema Execução-Geração- Partícula-Dimensão	60
Tabela 2 – Características básicas das funções de teste	64
Tabela 3 – Valores da calibração de C3 da PSO-CPS.....	65
Tabela 4 – Resultados Experimentais para as Funções de Teste.....	67
Tabela 5 – Taxa de Sucesso para as Funções de Teste.....	68
Tabela 6 – Comparação dos algoritmos CNPSO x PSO-CPS.....	77

1 - INTRODUÇÃO

A presente tese buscará inicialmente apresentar uma visão geral dos conhecimentos obtidos na Área Interdisciplinar de Computação de Alto Desempenho.

Conceitos relativos à otimização, soluções e complexidade dos problemas e seus métodos de resolução serão revisados para permitirem o estabelecimento de uma base para apresentação do contexto de desenvolvimento das soluções de problemas de otimização, suas divisões e classificações.

O detalhamento de técnicas de otimização e busca procura direcionar o estudo para o campo da solução dos problemas de tempo de execução não polinomial onde a utilização de soluções heurísticas tem sua eficiência efetivamente comprovada.

Propositamente, a revisão teórica será direcionada ao emprego de metaheurísticas, especificamente na área da computação evolucionária com a utilização do paradigma da Otimização por Enxame de Partículas e sua aplicação no presente trabalho.

1.1 - MOTIVAÇÃO

A literatura do desenvolvimento da metodologia de Otimização por Enxame de Partículas (PSO) tem apresentado diversas pesquisas de desempenho, desde a introdução do algoritmo padrão proposto por Kennedy e Eberhart, em 1995. Há uma constante tentativa de minimização dos problemas relativos ao processo de busca e à ocorrência de convergência prematura em mínimos locais.

São numerosos os estudos sobre a influência dos parâmetros utilizados, sobre características da estrutura da topologia de população bem como variações introduzidas no modelo original, de modo a aumentar o desempenho de busca do algoritmo. A partir da análise das evoluções propostas e com o foco na simplicidade de uma solução de melhoria de desempenho é que esse estudo tem sua motivação.

1.2 - OBJETIVO

A Otimização por Enxame de Partículas é um algoritmo de otimização baseado em população que tem obtido relevado reconhecimento na solução de problemas de grande dimensionalidade, com simplicidade e utilizando pequenos recursos computacionais.

O objetivo deste trabalho é propor a utilização de um mecanismo de comportamento biológico de grupos, denominado congregação passiva, com um processo seletivo, no modelo de otimização por enxame de partículas para aumentar o desempenho na solução de problemas de otimização global, combatendo o problema da convergência prematura pelo equilíbrio das fases de exploração global e local.

1.3 - ORGANIZAÇÃO DA TESE

Esta tese é composta, além deste capítulo, de outros seis capítulos adicionais que estão organizados da seguinte forma:

- O Capítulo 2 aborda o conceito de otimização como um processo de resolução de problemas e apresenta seus elementos básicos. A complexidade de algoritmos e classificações dos principais métodos de resolução de problemas são também revistos.
- O Capítulo 3 é dedicado à Computação Evolucionária, seu contexto histórico e suas principais características. Os algoritmos evolucionários são apresentados bem como suas vantagens e desvantagens em relação a outras técnicas. São exemplificadas as principais metodologias oriundas da computação evolucionária.
- O Capítulo 4 apresenta a Inteligência de Enxames e os conceitos utilizados na Otimização por Colônia de Formigas exemplificando os modelos iniciais desenvolvidos nessa metodologia.

- O Capítulo 5 apresenta metodologia de Otimização por Enxame de Partículas, suas características em relação aos algoritmos evolucionários. O histórico de elaboração do algoritmo da otimização é revisado e as principais melhorias são apresentadas.
- O Capítulo 6 apresenta a proposta do novo modelo de otimização por enxame de partículas com congregação passiva seletiva. Os conceitos utilizados, a metodologia da estrutura do algoritmo, bem como a análise e a apresentação da comparação de resultados com outras metodologias recentes complementam a proposta.
- O Capítulo 7 apresenta as conclusões e sugestões de trabalhos futuros.

2 - OTIMIZAÇÃO

O conceito de otimização é inerente à atividade humana. O homem desde as suas origens procura realizar suas tarefas ou necessidades de modo mais eficiente possível. A eficiência ao longo do tempo vem sendo avaliada de diversas formas, como por exemplo:

a) Avaliação Intuitiva de Sobrevivência

- aumento da capacidade de sobrevivência pela moradia em cavernas e convivência em grupos.

b) Avaliação Racional Econômica

- minimização do tempo de execução de uma tarefa.
- minimização do custo ou das perdas de uma tarefa.
- maximização dos lucros de uma tarefa.

O desenvolvimento humano é um contínuo gerador de complexidade nos problemas oriundos da escassez dos recursos e do aumento contínuo da demanda. Dessa forma, os problemas de otimização estão presentes no dia a dia em todas as atividades humanas, sejam elas sociais ou econômicas.

A otimização é uma ciência que está sempre em demanda por estar diretamente ou indiretamente envolvida na busca pela economia de recursos ou no aumento dos lucros de uma atividade.

Segundo SCHACHTER, 1990, a otimização é um processo de procura em se obter o melhor resultado possível para um problema, atendendo a determinados critérios e sujeito a restrições inerentes ao problema.

2.1 - ELEMENTOS DO PROCESSO DE OTIMIZAÇÃO

O processo de otimização é basicamente um processo de resolução de problemas englobando diversos elementos cujos conceitos são válidos em qualquer tipo de método de cálculo empregado:

- **Função Objetivo:** É a função que descreve o problema objeto da otimização. Para cada ponto do espaço solução do problema é atribuído um valor da função objetivo;
- **Variáveis Livres ou independentes:** São variáveis do espaço de solução que não dependem de nenhuma outra variável, podendo assumir qualquer valor. Cada ponto no espaço solução é uma combinação de variáveis livres;
- **Variáveis Dependentes:** São outras variáveis do problema, cujos valores dependem das variáveis livres;
- **Restrições:** São limitadores da viabilidade das soluções encontradas em função de características do problema;
- **Função penalidade:** Tem como finalidade quantificar uma restrição imposta a uma determinada condição de uma ou mais variáveis livres;
- **Espaço de Busca ou Espaço Solução:** É o conjunto de todas as soluções possíveis, viáveis e não viáveis, do problema. É definido pelo limite superior e inferior para cada variável livre do problema;
- **Ponto Ótimo:** É o ponto do espaço solução onde a função objetivo atinge o seu máximo (maximização) ou mínimo (minimização);
- **Ótimo local:** É o ponto máximo ou mínimo que ocorre em uma determinada região do espaço solução;

- **Ótimo Global:** É o ponto máximo ou mínimo que ocorre em toda a região do espaço solução.

2.2 - SOLUÇÕES E COMPLEXIDADE DE ALGORITMOS

Na busca de soluções para os problemas de otimização, a complexidade das soluções computacionais é bastante relevante, sendo necessária uma classificação dos algoritmos propostos de modo a determinar o tipo de solução a ser empregada.

Um dos métodos de análise da complexidade de um algoritmo é através de uma expressão matemática que retrate a ordem de crescimento do tempo de execução do algoritmo em função do tamanho da sua entrada. Dessa forma, a notação “ O ” é muitas vezes utilizada para exprimir complexidades (taxas de crescimento do tempo de execução) ou o limite superior mínimo do tempo de execução.

Segundo Cormen (CORMEN et al., 1990) os algoritmos podem ser classificados em relação à complexidade como:

- P (Deterministic Polynomial Time): são algoritmos cujo tempo de execução no pior caso não ultrapassa o teto $O(n^k)$, para alguma constante k , onde n é o número de entradas do problema.

- NP (Nondeterministic Polynomial Time): são algoritmos cujo tempo de execução tem como o teto $O(k^n)$, para alguma constante k , onde n é o número de entradas do problema.

Dentro do contexto da teoria da complexidade dos algoritmos enquadram-se os problemas ditos "NP-completos" e "NP-difícil". Por definição (NP-completos \subseteq NP-difícil). Na classe NP-difícil está a maioria dos problemas de otimização combinatória.

Problemas "NP" são uma classe especial de problemas de difícil solução de forma ótima, pois só podem ser resolvidos por algoritmos não-determinísticos polinomiais.

2.3 - MÉTODOS DE RESOLUÇÃO DE PROBLEMAS

De acordo com a viabilidade da busca pela solução ótima, normalmente função da quantidade de entradas do problema, podemos classificar os métodos de resolução em métodos exatos, métodos aproximativos e métodos heurísticos.

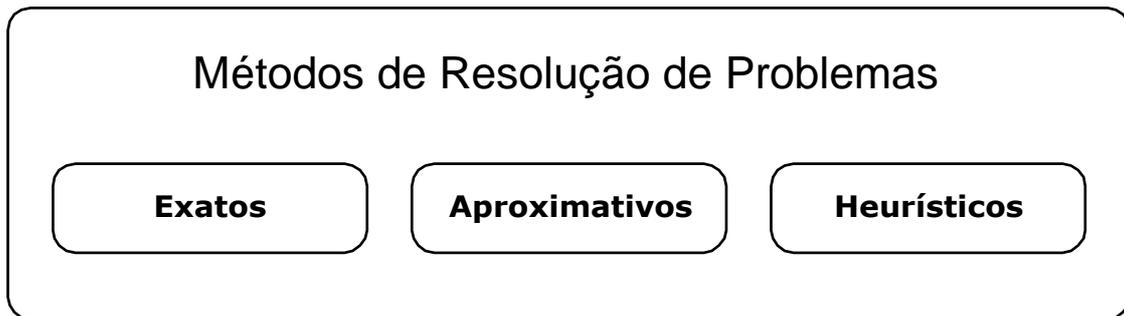


Figura 1 - Métodos de resolução de problemas

2.3.1 – Métodos Exatos

Os métodos de busca exatos são aqueles capazes de encontrar a solução ótima que satisfaz a função objetivo do problema em questão e seus parâmetros de controle e/ou restrições.

Nem sempre um problema pode apresentar um algoritmo exato para sua solução e há casos em que, mesmo existindo esse algoritmo, o mesmo pode requerer um tempo de processamento absurdo (meses, anos, décadas) para encontrar a solução. Quando isso ocorre diz-se que o problema é inviável para a instância abordada.

A escolha da forma de resolução dos problemas por métodos exatos deve sempre observar a viabilidade de execução do método escolhido.

Os métodos exatos baseiam-se em técnicas conhecidas de Programação Matemática e são normalmente viáveis para a resolução de problemas polinomiais, isto é, pertencentes à classe de problemas P.

As técnicas de Programação Matemática foram os primeiros métodos utilizados na resolução de problemas de otimização e por isso constituem a grande maioria dos livros de otimização [BAZARAA e SHETTY, 1979; HAFTKA e KAMAT, 1985; KIRSCH, 1981; LUENBERGER, 1984; PRESS et al., 1992].

Neves [NEVES, 1997] propõe, na figura 2, uma das classificações encontradas na literatura para os métodos clássicos (métodos matemáticos). Essa classificação é dividida inicialmente em:

- **Programação Linear:** é aquela em que a função objetivo e as restrições são funções lineares das variáveis do problema; e
- **Programação Não-Linear:** é aquela em que a função objetivo ou pelo menos uma das restrições é função não-linear das variáveis do problema.

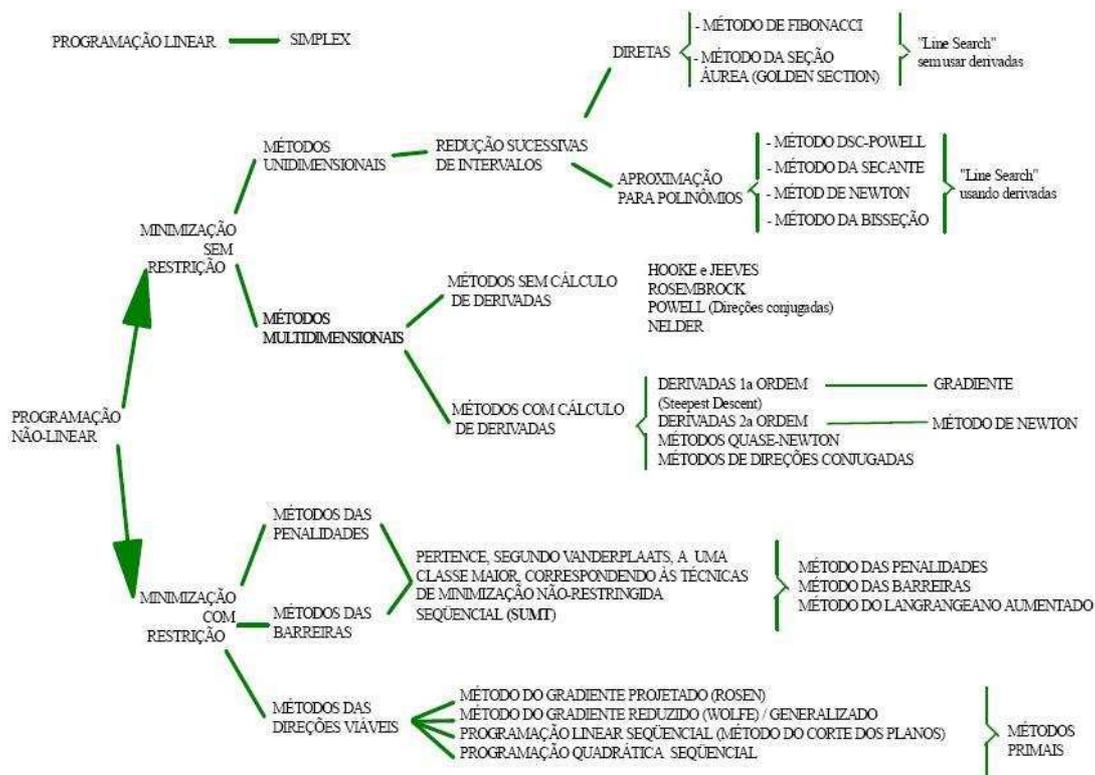


Figura 2 - Classificação Geral dos Métodos Clássicos [NEVES,1997]

2.3.2 – Métodos Aproximativos

Os métodos aproximativos são utilizados quando se deseja encontrar soluções para problemas exponenciais onde os métodos exatos não se aplicam computacionalmente. Segundo Cormen [CORMEN et al., 1990] a utilização de algoritmos de aproximação é viável, pois os mesmos, por possuírem tempo polinomial, podem resolver problemas exponenciais com boa margem de aproximação à solução ótima.

Os métodos aproximativos normalmente garantem que a solução se aproxima da solução ótima a cada iteração. Entretanto, a rapidez na convergência pode ser questionada em relação a outros métodos.

Ainda, segundo Cormen (CORMEN et al., 1990), vários problemas clássicos da literatura possuem soluções aproximativas como alternativa à demanda computacional requerida por métodos exatos em problemas NP. Dentre eles, destacam-se:

- Problema de Cobertura de Vértices (Vertex Cover Problem);
- Problema do Caixeiro Viajante (Travelling Salesman Problem);
- Problema da Cobertura de Conjuntos (Set Covering Problem); e
- Problema da Soma de Subconjuntos (Subset Sum Problem).

2.3.3 – Métodos Heurísticos

Os métodos heurísticos podem ser considerados como uma nova forma de abordagem para solução de problemas de otimização combinatória onde o objetivo é a procura por soluções “boas” (as melhores possíveis) que não exijam um esforço computacional excessivo, especialmente para aplicações práticas.

Heurísticas são algoritmos que geram soluções viáveis para as quais não se pode dar garantias de qualidade, isto é, os métodos heurísticos não garantem qualquer tipo de solução conforme o número de iterações, além de não haver garantia de convergência, ou seja, não se sabe o quão distante a solução gerada está da solução ótima. Entretanto,

na prática as heurísticas costumam convergir em tempo extremamente rápido quando comparadas aos métodos aproximativos.

Tipos de heurísticas:

- Heurísticas Construtivas: Normalmente adotam estratégias gulosas para construir soluções. Tipicamente são aplicadas a problemas onde é fácil obter uma solução viável.

- Heurísticas de Busca Local: Partem de uma solução inicial e, através de transformações bem definidas, visitam outras soluções até atingir um critério de parada pré-definido.

Conceitualmente, quando um método é aplicado especificamente a um tipo de problema, sob a forma de um algoritmo, este é denominado um método heurístico, ou simplesmente uma heurística. Como exemplo de heurística pode-se citar:

Sistema de Colônia de Formigas (AntSystem) [DORIGO, 1991], heurística aplicada à resolução do Problema do Caixeiro Viajante (Travelling Salesman Problem).

Quando um método é capaz de resolver vários tipos de problemas de otimização combinatória, o mesmo é denominado metaheurística. Além da abrangência de resolução dos problemas, as metaheurísticas possuem como principal característica a existência de mecanismos internos que evitem parar em um ótimo local (mesmo que para isso piorem a valor da função objetivo). Como principais metaheurísticas conhecidas podemos citar:

- Algoritmos Genéticos (Genetic Algorithm, GA) [HOLLAND, 1975];

- Otimização por Colônia de Formigas (Ant Colony Optimization, ACO) [DORIGO et al., 1999];

- Otimização por Enxame de Partículas (Particle Swarm Optimization, PSO) [EBEHART & KENNEDY, 1995];

- Busca Tabu (Tabu Search) [GLOVER, 1989-1990]);
- Simulated Annealing [KIRKPATRICK et al., 1985]; e
- GRASP (Greedy Randomized Adaptive Search Procedures) [FEO & RESENDE, 1995].

Recentemente, Dréo [DRÉO, 2007] propôs uma classificação das metaheurísticas baseada nas seguintes classes:

- metaheurísticas versus busca local;
- baseadas em população versus baseada em trajetória;
- computação evolucionária ou não;
- métodos inspirados na natureza ou não;
- função objetivo dinâmica ou estática;
- algoritmos baseados em memória ou não; e
- metaheurísticas implícitas, explícitas ou diretas (função densidade de probabilidade da amostra).

Ainda segundo Dréo [DRÉO & SIARRY, 2007], as classes referentes à função densidade de probabilidade da amostra referem-se às seguintes condições:

- a) Implícitas: Uma função densidade de probabilidade (FDP) é usada para categorizar a amostra (Algoritmos Evolucionários);
- b) Explícitas: Uma específica FDP é usada (Algoritmos de Estimativa de Distribuição); e
- c) Diretas: Uma aproximação da função objetivo é usada como uma FDP da amostra (Recozimento simulado).

A figura 3 apresenta uma classificação das metaheurísticas, modificada em relação à proposta inicial de Dréo [DRÉO, 2007]. Na classe relativa às metaheurísticas definidas como computação evolucionária foram incluídas a Otimização por Colônia de Formigas (Ant Colony Optimization, ACO) [DORIGO, 1999] e a Otimização por

Enxame de Partículas (Particle Swarm Optimization, PSO) [EBEHART & KENNEDY, 1995]. Essa alteração está de acordo com o conceito de computação evolucionária da maioria dos autores. Foi incluída/corrigida também a metodologia da programação evolucionária, criada por L. J. Fogel (1962) e desenvolvida por Bäck [BÄCK et al., 1997].

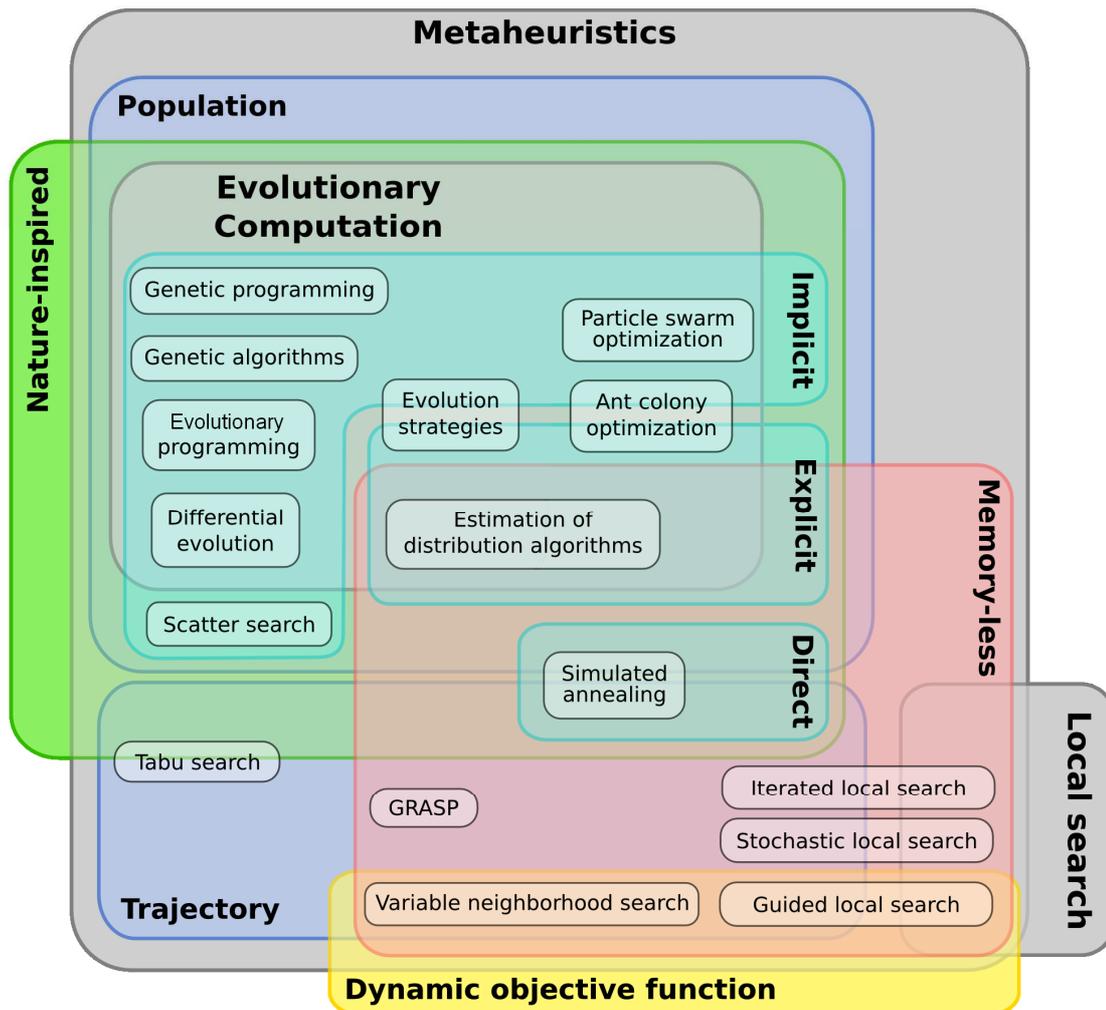


Figura 3 - Classificação das Metaheurísticas [DRÉO, 2007], modificada pelo autor dessa tese.

3 - COMPUTAÇÃO EVOLUCIONÁRIA

Segundo Mendes [MENDES, 2004], a Computação Evolucionária (CE) é um termo recente que engloba paradigmas de solução de problemas cujo elemento chave é a simulação de algum tipo de mecanismo de evolução conhecido.

Na computação evolucionária os modelos são inspirados em paradigmas biológicos observados na natureza. Os conceitos da evolução natural como reprodução, seleção e mutação, presentes na genética, bem como os mecanismos de cooperação e competição do comportamento coletivo dos indivíduos de uma população, foram observados e utilizados na geração de modelos matemáticos simplificados da realidade. Esses modelos matemáticos incorporam conceitos simples de evolução de soluções e são largamente aplicáveis à resolução de problemas em várias áreas do conhecimento.

A computação evolucionária possui a característica singular da simplicidade dos métodos de otimização observados na natureza e, do ponto de vista prático, consegue resolver problemas de modelos matemáticos complexos de modo simples.

A utilização de um conjunto de soluções, gerado iterativamente, cuja evolução baseia-se em processos modelados a partir de métodos de otimização observados na natureza, é avaliado continuamente e as soluções encontradas estão próximas à solução ótima de um problema.

Segundo Fogel [FOGEL, 1994, 1997], os primeiros estudos da CE datam dos anos 50; entretanto, por quase três décadas esse conhecimento ficou inexplorado pela comunidade científica por dois fatores principais: inexistência de capacidade computacional à época e formalização e caracterização deficientes das metodologias evolutivas nos primeiros estudos nessa área.

Para Kennedy e Eberhart [KENNEDY & EBERHART, 2001], a CE difere dos paradigmas tradicionais de busca e otimização em três aspectos:

- 1 – A utilização de uma população de soluções potenciais;

2 – O processo de decisão é guiado unicamente pela informação de aptidão;

3 – As regras de transição são mais probabilísticas do que determinísticas.

Outra característica da CE é a possibilidade de utilização de codificação de parâmetros em formas binárias e outros símbolos, permitindo assim uma melhor manipulação dos parâmetros.

3.1 - ALGORITMOS EVOLUTIVOS

Os algoritmos evolutivos (AE) são as metodologias da área de computação evolucionária, estabelecidos a partir de princípios existentes no “mundo biológico”, descritos nos princípios da evolução e comportamento social das espécies Darwiniana [DARWIN, 1859].

De acordo com Coelho [COELHO & COELHO, 1999], na década de 70 observou-se um número relevante de publicações e conferências nas áreas de CE. A maioria das abordagens correntes de AE descende dos princípios de diferentes metodologias desenvolvidas nas décadas de 50 a 70, como por exemplo:

- a) Algoritmos Genéticos: desenvolvidos principalmente por A. S. Fraser, H. J. Bremermann, J. Reed e J. H. Holland, entre a década de 50 e 70, com refinamentos posteriores por D. Whitley, D. E. Goldberg, K. De Jong e J. Grefenstette;
- b) Programação Evolutiva: desenvolvidas por L. J. Fogel, A. J. Owens e M. J. Walsh, nos Estados Unidos, na década de 60, refinada recentemente por D. B. Fogel, G. H. Burgin, P. J. Angeline, V. W. Porto e W. Atmar;
- c) Estratégias Evolutivas: desenvolvidas na Alemanha, por I. Rechenberg e H.P. Schwefel, na década de 60, com aprimoramentos posteriores de G. Rudolph, H. G. Beyer, F. Kursawe e T. Bäck; e

- d) Programação Genética, abordadas pelos pesquisadores J. R. Koza, J. P. Rice, K. E. Kinnear e P. J. Angeline.

Entre os anos de 80 e 90, com o avanço no desempenho das plataformas computacionais, a aplicação dos AE pôde ser definitivamente desenvolvida e empregada na resolução de problemas de otimização do mundo real. Atualmente, as áreas de aplicação dos AE abrangem: engenharia, telecomunicações, escalonamentos, robótica móvel, manufatura, reconhecimento de imagens, identificação, previsão, controle, processamento de imagens, processamento de sinais.

O sucesso dos AE baseia-se na possibilidade de soluções para problemas altamente complexos utilizando-se como sugestão os exemplos de soluções de problemas observados na natureza. Esses algoritmos são especialmente úteis nas tarefas de otimização global e busca, onde os algoritmos determinísticos podem levar a soluções de mínimos locais. Os AE são aptos à resolução de uma vasta gama de problemas não-lineares, descontínuos, discretos e multivariáveis.

Segundo Coelho [COELHO & COELHO, 1999], o emprego de AE na configuração de algoritmos de otimização e aprendizado de máquina (machine learning) é empregado com sucesso devido às seguintes características:

- (i) tratam adequadamente os sistemas sujeitos a restrições;
- (ii) não requerem as informações relativas a derivadas, estas usualmente necessárias em métodos convencionais de otimização;
- (iii) adequam-se à implementação em paralelo e distribuídas;
- (iv) possibilitam a utilização do conhecimento obtido a priori pelo projetista;
- e
- (v) tratam sistemas complexos e espaços de busca com múltiplas modas e/ou múltiplos objetivos.

Independente do paradigma implementado, o ciclo básico dos dados num AE pode ser sintetizado nos seguintes passos:

- (i) inicialização aleatória da população de soluções;

- (ii) avaliação da função de aptidão;
- (iii) seleção dos indivíduos mais aptos de acordo com uma estratégia de seleção;
- (iv) aplicação dos operadores de recombinação e mutação;
- (v) geração de uma nova população de soluções candidatas;
- (vi) repetição dos passos (ii) a (v) até que uma condição de parada seja satisfeita.

Segundo Goldberg [GOLDBERG, 1989] um AE possui as seguintes características:

- (i) operar em uma população de pontos;
- (ii) não requerer cálculos de derivadas e informação sobre o gradiente da função objetivo;
- (iii) trabalhar com a codificação de seu conjunto de parâmetros, não com os próprios parâmetros (representação binária);
- (iv) realizar transições probabilísticas, em vez de regras determinísticas;
- (v) necessitar apenas da informação sobre o valor da função objetivo para cada indivíduo da população;
- (vi) apresentar simplicidade conceitual; e
- (vii) ser pouco afetado, quanto à eficiência, quando descontinuidades e ruídos estão presentes nos dados do problema.

As características (iii) a (v) não são comuns a todos os AE, mas geralmente presentes nos algoritmos genéticos.

Nos algoritmos evolutivos o princípio básico do funcionamento é a execução de um processo evolutivo sobre uma população inicial, representada por um conjunto de indivíduos que representam possíveis soluções do problema. Durante processo evolutivo, geralmente baseado em uma analogia com algum processo natural, cada indivíduo da população é avaliado, segundo uma função objetivo ou função de aptidão e as características das melhores soluções são preservadas e utilizadas como um novo conjunto de soluções tentativas que tendem a se aproximar da solução ótima do problema.

Normalmente o processo evolutivo é interrompido por um critério de parada definido quando um número de evoluções pré-determinado tenha ocorrido ou quando é atingido um determinado desempenho desejado.

Para permitir a não convergência das soluções para uma solução subótima (ponto ótimo local) os algoritmos evolutivos empregam técnicas de aleatoriedade na geração de soluções tentativas de modo a apresentarem novas características ao conjunto de soluções. Dessa forma um certo grau de aleatoriedade no processo evolutivo normalmente tende a explorar melhor o espaço de soluções do problema.

Os AE apresentam vantagens e desvantagens em relação aos métodos tradicionais de busca e otimização [COELHO, 2003].

Entre as vantagens dos AE tem-se:

- (i) não existe a necessidade de assumir-se características do espaço do problema;
- (ii) vastamente aplicável (algoritmos de propósito geral);
- (iii) baixo custo de desenvolvimento e aplicação;
- (iv) facilidade de incorporar outros métodos; e
- (v) pode ser executado interativamente e possibilita a acomodação de soluções propostas pelo usuário no procedimento de otimização.

Entre as desvantagens dos AE deve-se mencionar que:

- (i) não garantem uma solução ótima;
- (ii) podem necessitar de sintonia de alguns parâmetros inerentes à metodologia evolutiva adotada; e
- (iii) tratam-se de métodos estocásticos e seu desempenho varia de execução para execução (a menos que o mesmo gerador de números aleatórios com a mesma semente seja utilizado).

3.1.1 – Algoritmos Genéticos

Os algoritmos genéticos (AG) são técnicas não-determinísticas de busca e otimização que operam uma população de possíveis soluções utilizando processos inspirados na seleção natural de Darwin e da genética de Mendel.

Conceitualmente o AG é o mais conhecido paradigma da computação evolucionária. Em um AG a solução de um problema é um cromossomo individual de uma população de indivíduos que tenta sobreviver sob severas condições. A habilidade de sobreviver é representada pela aptidão, que permite que os indivíduos possam se reproduzir e gerar descendentes.

O princípio básico do funcionamento dos Algoritmos Genéticos é o estabelecimento de um processo evolutivo executado sobre uma população inicial, formada por um conjunto de indivíduos que podem ser vistos como possíveis soluções do problema. O processo evolutivo, também denominado reprodução, congrega diversas fases onde, basicamente, cada indivíduo da população é avaliado, segundo uma função de aptidão ou função objetivo, para refletir sua habilidade de adaptação a um determinado ambiente. Uma porcentagem dos indivíduos mais adaptados é mantida, enquanto os outros são eliminados (seleção natural). Os indivíduos selecionados podem sofrer uma série de modificações em suas características, como mutações e cruzamentos, ou recombinação genética como forma de geração de descendentes para a próxima população. Nos Algoritmos Genéticos o processo seleção e modificações de características dos indivíduos selecionados de uma população é implementado por operadores genéticos.

Os AG geralmente usam três operadores: seleção, recombinação (crossover) e mutação. Cada operador empregado no AG representa uma função existente no processo de evolução como: sobrevivência dos indivíduos mais aptos (seleção), reprodução de melhores indivíduos (recombinação) ou mesmo restauração de material genético perdido (mutação).

Algoritmos Genéticos são muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas em problemas com as seguintes características básicas:

- Problemas complexos de otimização;
- Grande quantidade de parâmetros e variáveis;
- Problemas mal estruturados com condições e restrições difíceis de serem modelados matematicamente;
- Grandes espaços de busca; e
- Objetivos Múltiplos.

3.1.2 – Programação Evolucionária

A programação evolucionária (PE) é uma técnica semelhante aos AG, pois também utiliza uma população de possíveis soluções de um específico problema. A diferença básica da PE está na ênfase à ligação comportamental (relação fenotípica) entre as populações geradas em contrapartida à relação genética observada nos AG.

Segundo Fogel [FOGEL, 1994] os algoritmos genéticos trabalham no espaço genótipo enquanto a programação evolucionária enfatiza o espaço fenótipo.

O ciclo básico de um algoritmo de PE pode ser sintetizado nos seguintes passos:

- (i) inicialize a população.
- (ii) exponha a população ao ambiente.
- (iii) calcule a aptidão de cada membro.
- (iv) randomicamente faça a mutação de cada “pai” membro da população.
- (v) avalie pais e filhos.
- (vi) selecione membros da nova população
- (vii) repetição dos passos (ii) a (vi) até que uma condição de parada seja satisfeita.

3.1.3 – Estratégias de Evolução

Se a programação evolucionária é baseada em evolução, então segundo Rechenberg [RECHENBERG, 1994] o campo da estratégia de evolução (EE) é baseado na evolução da evolução. Se o processo biológico foi otimizado por um processo de evolução e a evolução é um processo biológico então o processo de evolução também deve ser otimizado.

O ciclo básico de um algoritmo de PE pode ser sintetizado nos seguintes passos:

- (i) inicialize a população.
- (ii) execute a recombinação usando μ pais para gerar λ filhos.
- (iii) execute a mutação de todos os filhos.
- (iv) avalie λ ou $\mu + \lambda$.
- (v) selecione μ para a nova população.
- (vi) repetição dos passos (ii) a (v) até que uma condição de parada seja satisfeita.
- (vii) repetição dos passos (ii) a (v) até que uma condição de parada seja satisfeita.

3.1.4 – Programação Genética

A maioria dos algoritmos evolucionários geralmente manipula estruturas individuais definidas como cadeias ou vetores de valores binários ou variáveis reais. O paradigma da programação genética (PG) manipula programas de computadores e geralmente variam de tamanho, forma e complexidade. Existem três formas de representação de programas de computadores: árvore, grafo e representação linear. A representação em árvore é a mais utilizada.

As diferenças básicas entre um GP e um AG são:

- Os membros da população são estruturas executáveis (geralmente programas de computadores); e

- A aptidão de um membro individual da população é medida pela execução do mesmo.

O objetivo de uma PG é descobrir que programa de computador fornece uma desejada saída para um dado conjunto de entrada.

Cada programa é geralmente representado como uma árvore, onde as funções aparecem nos ramos e as variáveis e constantes nas folhas. Segundo Kennedy e Eberhart [KENNEDY & EBERHART, 2001] em PG para achar o programa que resolve um determinado problema é necessário especificar:

- (i) o conjunto terminal (variáveis e constantes do problema).
- (ii) o conjunto função (funções matemáticas, operadores aritméticos, operadores booleanos, operadores condicionais, funções iterativas e recursivas).
- (iii) a medida de aptidão.
- (iv) o sistema de controle de parâmetros (tamanho da população, probabilidade de reprodução, probabilidade de recombinação, tamanho máximo hierárquico).
- (v) Especificação das condições de término (número máximo de gerações).

4 - INTELIGÊNCIA DE ENXAMES

Segundo Bonabeau [BONABEAU et al., 1999], o termo inteligência de enxames foi usado originalmente para descrever um particular paradigma na pesquisa de robôs. Ainda nesse texto os autores ampliaram o conceito de inteligência de enxames para incluir qualquer tipo de algoritmo ou esquema de solução de problemas inspirado no ambiente coletivo de colônia de insetos ou outras sociedades de animais.

Para Kennedy e Eberhart [KENNEDY & EBERHART, 2001], o termo enxames (swarm) deve ser ainda menos restritivo, devendo ser atribuído a qualquer tipo e estrutura fechada de interação de agentes. O exemplo clássico de enxame é o enxame de

abelhas, mas a metáfora de um enxame pode ser estendida para outros sistemas como arquitetura similar. Uma colônia de formigas pode ser entendida como um enxame cujos agentes individuais são formigas, um bando de pássaros são enxames onde os agentes são pássaros, o tráfego um enxame de carros, uma multidão um enxame de pessoas, um sistema imune um enxame de células e moléculas e uma economia um enxame de agentes econômicos.

4.1 - OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

A concepção inicial dos algoritmos inspirados em colônia de formigas tem sua origem em uma experiência executada por Deneubourg [DENEUBOURG et al., 1989] usando uma colônia de formigas argentinas reais (*Iridomyrmex humilis*). A experiência consistia na observação do comportamento das formigas na procura por alimento utilizando-se para isso de uma ponte dupla, com ramos de comprimento variável, de ligação entre o ninho e o alimento. Após uma fase inicial transitória, que pode durar poucos minutos, observa-se que as formigas tendem a convergir para o caminho mais curto. O aparecimento desse caminho mais curto pode ser explicado em termos de autocatálise ou processo de realimentação positiva (introdução de uma substância química denominada catalisador).

De fato, as formigas, enquanto vão do ninho para a fonte de comida e vice-versa depositam no solo uma substância química denominada feromônio, possibilitando uma forma de comunicação indireta por modificação local do ambiente (estigmergia). Ao longo do processo os feromônios são depositados em concentrações mais altas no caminho mais curto. Dessa forma quando as formigas ao depararem com um ponto de decisão (interseção entre dois caminhos) elas fazem uma escolha probabilística baseada na quantidade de feromônio que as mesmas sentem pelo cheiro.

Após alguns percursos das formigas surge então uma rede de trilhas de feromônios construída de forma auto-organizada e baseada em realimentação positiva. Este fenômeno adaptativo e robusto é um dos melhores exemplos de sistemas de inteligência coletiva.

As trilhas de feromônios são um tipo de informação numérica distribuída que é modificada pelas formigas para refletir sua experiência quando da resolução de um problema em particular. Dorigo [DORIGO et al., 1991] e Manierzo [MANIERZO & COLORNI, 1999].

4.1.1 – Modelo Ant System

O modelo Ant System criado por Dorigo [DORIGO et al., 1991] é um exemplo inicial de modelo de otimização estocástica baseado na utilização de agentes simples, denominadas formigas artificiais.

Originalmente o modelo foi aplicado ao clássico Problema do Caixeiro Viajante (Travelling Salesman Problem - TSP) onde resumidamente procura-se determinar o menor caminho a ser percorrido pelo caixeiro viajante em um caminho composto por cidades conectadas por estradas. O caminho deve sair de uma cidade inicial e visitar todas as outras cidades apenas uma vez e retornar à cidade inicial. De acordo com correlação com os grafos, os nós serão entendidos como cidades e as arestas como as estradas do TSP. O caminho completo passando por todos os nós será definido como um *tour*.

A implementação adotada implica que os agentes formiga escolham as arestas probabilisticamente no *tour*, em função do tamanho e da quantidade de feromônio nas arestas. A evaporação e a reinserção de feromônio só são realizadas após todas as formigas terem completado o *tour* em uma relação inversa à distância total percorrida, ou seja, as formigas que fizeram o caminho mais curto têm direito de inserir mais feromônio nas arestas.

4.1.2 – Modelo Ant Colony System

O modelo Ant Colony System [DORIGO et al., 1997] utiliza como base o modelo Ant System, descrito no item 4.1.1. As principais diferenças em relação ao modelo anterior permitiram a transformação do modelo Ant System em uma metaheurística reconhecidamente capaz de resolver problemas de otimização combinatória em geral, tais como: o problema do caixeiro viajante [DORIGO et al.,

1997], ordenação seqüencial [GAMBARDELLA & DORIGO, 1997], especificação quadrática [GAMBARDELLA et al., 1997], (MANIEZZO & COLORNI, 1999) e problemas de roteamento de veículos (BULLNHEIMER et al., 1998), (GAMBARDELLA et al., 1999).

São as seguintes as principais diferenças do modelo Ant Colony System:

a) A introdução de uma nova equação, denominada *pseudo-random-proportional* na regra de transição. No início do algoritmo deverá ser escolhido um parâmetro q_0 ($[0,1]$). A cada decisão da formiga em relação à escolha de uma aresta deverá ser sorteado um valor q entre 0 e 1. Se este valor for maior que q_0 , a aresta escolhida é aquela que tiver maior probabilidade de escolha (*exploitation*), caso contrário, deve-se utilizar a equação de transição definida em Ant System (Figura 8) (*biased exploration*) onde torna-se necessário multiplicar as probabilidades por um número randômico para se chegar ao maior valor;

b) A modificação da regra de atualização global, de forma que apenas a formiga que percorreu o menor caminho global possa atualizar o feromônio; e

c) A inserção da atualização local do feromônio, com a atualização do feromônio durante a realização do *tour*.

5 - OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

O paradigma da otimização por enxame de partículas (Particle Swarm Optimization) é uma heurística moderna cuja origem encontra-se nos processos sócio-cognitivos observados e empregados em modelos abstratos da inteligência e aprendizado coletivo.

A proposta original de uma simulação de um ambiente social foi inicialmente introduzida como método de otimização em 1995, por Kennedy e Eberhart [KENENDY & EBERHART, 1995] e [EBERHART & KENENDY, 1995]. Particle Swarm Optimization (PSO) está relacionada com vida artificial, especificamente teoria de enxames, e também com computação evolucionária, especialmente com estratégias de evolução e algoritmos genéticos.

Segundo Eberhart [EBERHART et al., 1996] a PSO é facilmente implementável e seu custo computacional é inexpressivo desde que os requisitos de memória e velocidade de CPU estejam baixos.

Para Kennedy e Eberhart [KENENDY & EBERHART, 2001] o método PSO pertence à categoria de métodos de inteligência de enxames. PSO também tem provado ser um eficiente método quando empregado na solução de problemas de otimização global e em alguns casos não experimenta as dificuldades encontradas por outras técnicas da computação evolucionária [EBERHART & KENENDY, 1995].

5.1 - ENXAME DE PARTÍCULAS É UM TIPO DE ALGORITMO EVOLUCIONÁRIO?

Kennedy e Eberhart [KENNEDY & EBERHART, 2001] em seu livro “Swarm Intelligence” colocam em discussão se o enxame de partículas é realmente um tipo de algoritmo evolucionário (AE).

Os algoritmos de enxame de partículas têm sido influenciados pelos conceitos da computação evolucionária desde o seu início e continuam a ser. O uso de populações de soluções individuais do problema provém diretamente das tradições da CE e aspectos do

algoritmo de enxame de partículas podem ser vistos de modo semelhante aos operadores evolucionários. A comunidade de CE tem recebido o enxame de partículas como um modo alternativo de fazer as coisas, e os pesquisadores de enxame de partículas consideram-se co-participantes no processo de reconhecimento de o exame de partículas ser um tipo de algoritmo evolucionário.

A evolução de um organismo é determinada não somente pelas pressões do processo de seleção, mas também por um conjunto de restrições e oportunidades oferecidas ao longo do seu desenvolvimento. Essa nova percepção da evolução surge com a seleção natural e a auto-organização trabalhando em conjunto, isto é:

$$\text{evolução} = \text{seleção natural} + \text{auto-organização}$$

Considerações sobre o modo de operação de busca dos algoritmos de enxame de partículas em relação aos métodos tradicionais de computação evolucionária destacam que as diferenças entre essa técnicas são consideráveis. Entretanto, operações evolucionárias são notadamente análogas aos métodos do enxame de partículas e dessa forma Kennedy e Eberhart [KENNEDY & EBERHART, 2001] concluem que os algoritmos evolucionários e algoritmos sociais têm muito em comum e que sua fusão parece ser inevitável.

5.2 - HISTÓRICO

As regras implícitas, seguidas por membros de bando de pássaros e cardumes de peixes, que os habilitam a moverem-se de forma sincronizada, sem colisão, resultando uma coreografia surpreendente, foi estudado e simulado por vários cientistas [PARSOPOULOS & VRAHATIS, 2002].

Nas simulações, o movimento das “coleções” foi resultante do comportamento dos seus indivíduos (pássaros, peixes, etc.) nos seus esforços para manter um distância ótima de cada um de seus vizinhos individuais [EBERHART et al., 1996].

Entre os estudos dedicados a comportamento de “coleções” destaca-se o de Reynolds [REYNOLDS, 1987] onde um algoritmo foi elaborado assumindo que um bando de pássaros quando voam em conjunto baseiam-se em três regras:

- evitar a colisão com seus vizinhos.
- manter a velocidade de seus vizinhos
- tentar mover-se próximo ao centro do bando por observação

Os resultados apresentavam coerentes clusters de pássaros simulados rodopiando através do espaço. Esse algoritmo foi empregado em várias sequências cinemáticas de diversas coleções.

Segundo Grosnan [GROSNAN, ABRAHAM, CHIS, 2006], quando um grupo de agentes coopera para atingir um propósito ou algum objetivo comum, um tipo de auto-organização emerge das ações realizadas por esse grupo. Essa inteligência coletiva aparece frequentemente em grandes grupos ou coleções de agentes simples como bandos de pássaros, cardumes de peixes, enxames de insetos. Os agentes usam simples regras locais para governar suas ações e via interações com os demais membros do grupo, a coleção de agentes atinge seus objetivos.

Segundo Fayyad [FAYYAD et al., 1996], um agente autônomo é um subsistema que interage com seu ambiente, que provavelmente consiste de outros agentes, mas atua relativamente independente de outros agentes. O agente autônomo não segue os comandos de um líder, ou algum plano global. Por exemplo, para um pássaro participar de um bando, ele somente ajusta seus movimentos para coordená-los com o movimento do bando, tipicamente como se comportam seus vizinhos mais próximos. Um pássaro em um bando simplesmente tenta evitar a colisão com seus vizinhos. Um ambiente de bando ajuda os pássaros a terem vantagem sobre diversos aspectos incluindo proteção contra predadores (especialmente para pássaros no meio do bando), busca de alimento (essencialmente cada pássaro tem sua visão expandida pelos olhos de cada um dos demais pássaros do bando).

Existem numerosos exemplos na natureza que reforçam a idéia de que o compartilhamento de informação entre os indivíduos de uma população possa trazer

uma vantagem evolucionária. Esta é a idéia central do desenvolvimento da PSO [EBERHART et al., 1996].

A origem do PSO foi um simulador simplificado de um sistema social, que foi usado para visualizar o movimento de um bando de pássaros. Diversas versões desse simulador foram desenvolvidas e testadas até que a primeira versão simples do PSO foi a apresentada por Eberhart, Kennedy e outros [EBERHART & KENNEDY, 1995] e [EBERHART et al., 1996].

5.3 - O ALGORITMO DA OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

PSO é similar às técnicas da computação evolucionária onde o uso de uma população de potenciais soluções do problema em questão é utilizada para investigar o espaço de busca. Outra característica da CE é a utilização de uma função de aptidão que permite a quantificação da qualidade individual de uma possível solução.

Cada indivíduo de um enxame é denominado partícula e a mesma é composta de três vetores de armazenamento cuja dimensionalidade é dada pela própria dimensionalidade do problema. Os vetores são:

- posição \mathbf{X}_i
- melhor posição \mathbf{P}_i
- velocidade \mathbf{V}_i

A posição \mathbf{X}_i representa um conjunto de coordenadas Cartesianas descrevendo um ponto do espaço de solução.

A cada iteração, a posição corrente é avaliada como uma solução do problema. Se a solução encontrada é melhor do que uma outra solução já encontrada anteriormente (\mathbf{P}_i) então ela substituirá o valor armazenado em \mathbf{P}_i .

A aptidão do melhor resultado encontrado de cada partícula é armazenada em \mathbf{PBEST}_i para simplificar comparações com iterações futuras.

Cada partícula é membro de uma vizinhança social, e a influência social é resultado da observação de seus vizinhos. Nesse sentido a partícula será afetada pelo melhor ponto encontrado por qualquer membro da sua vizinhança topológica. A melhor posição encontrada pelos indivíduos com os quais a partícula mantém ligações é armazenada em **GBEST**.

Então:

PBEST – melhor posição individual

GBEST – melhor posição social

A definição da topologia de uma vizinhança é simplesmente a caracterização de uma rede social, representado como um grafo, onde cada indivíduo é representado como um vértice e um eixo existente entre dois indivíduos se eles podem influenciar um ao outro. Vizinhos e vizinhança não se alteram durante a corrida.

Essa rede de comunicação no enxame que determina as inter-relações entre as partículas e as relações de cada partícula com todas as outras partículas é conhecida como Sociometria do Enxame.

Existe uma infinita quantidade de topologias de vizinhança. Entretanto, somente dois tipos têm sido frequentemente usados. Na topologia **GBEST** (melhor global) cada partícula influencia todas as outras; é o caso da conexão total entre os indivíduos. Na topologia **LBEST** (melhor local) cada partícula está conectada com **K** vizinhos adjacentes, onde **K** é um parâmetro. Muitas aplicações incluem a própria partícula **i** como um membro da sua vizinhança; então, a vizinhança de **k = 2** na realidade possui 3 membros.

A figura 4 apresenta a representação gráfica das topologias **GBEST** e **LBEST**.

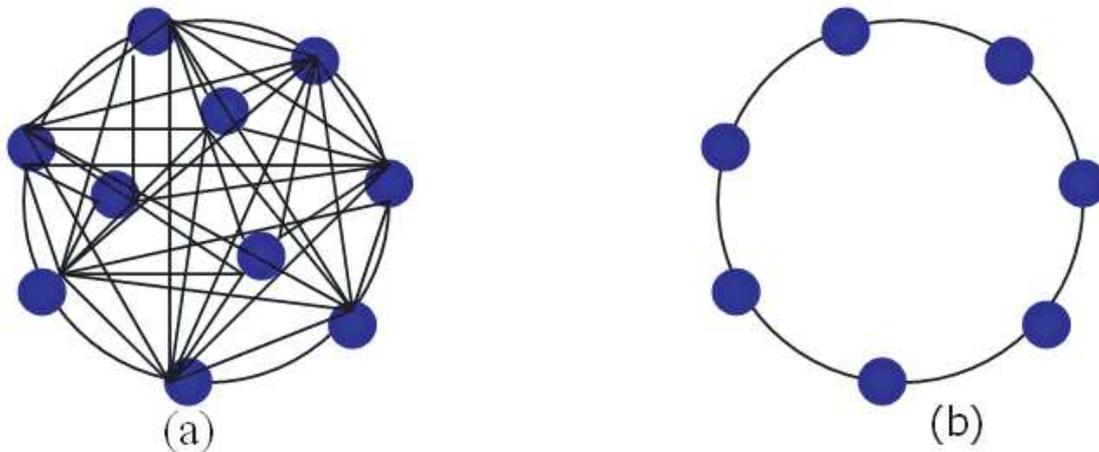


Figura 4 – Representação gráfica de (a) GBEST e (b) LBEST

Na topologia GBEST a trajetória de cada partícula individual é influenciada pela melhor posição individual de toda a população. Assume-se que em GBEST o algoritmo converge rápido, já que todas as partículas são atraídas simultaneamente para a melhor parte do espaço de busca. Entretanto, se o ótimo global não está perto da melhor partícula, pode ser impossível que as partículas explorem outras áreas e, conseqüentemente, podem ficar presas a um ótimo local [KENNEDY & MENDES, 2002].

Na topologia LBEST, cada partícula é influenciada por um pequeno número de seus vizinhos (que podem ser vistos como membros da vizinhança adjacente da partícula). Normalmente, a vizinhança LBEST compreende dois vizinhos, um do lado esquerdo e outro do lado direito. Este tipo de topologia converge devagar, mas pode localizar um ótimo global com uma grande chance. Assume-se que em LBEST as partículas circulem por diferentes ótimos locais [KENNEDY & MENDES, 2002].

No algoritmo PSO a evolução da população é realizada através do “vôo” de cada partícula pelo espaço de busca. Dessa forma o conceito de gerações é substituído pelo conceito de intervalos de tempo.

A evolução das partículas é a sua “velocidade de vôo” que é definida como um vetor dentro do espaço de busca. A cada intervalo de tempo a velocidade é atualizada e uma nova posição da partícula é calculada em função da posição atual e da nova velocidade calculada.

A equação de atualização da velocidade possui três componentes. O primeiro é um componente inercial, função da velocidade prévia da partícula e que tende a manter a velocidade inalterada. O segundo componente é o componente histórico individual e tende a levar a partícula para o melhor ponto já visitado pela mesma. O terceiro componente é denominado componente histórico social e tende a levar a partícula para o melhor ponto já visitado pelo enxame.

O algoritmo básico inicial do PSO proposto por Kennedy [KENNEDY & EBERHART, 1995] é composto de duas fórmulas:

$$V_{i+1} = V_i + 2 * \text{rnd}() * (\text{PBEST} - X_i) + 2 * \text{rnd}() * (\text{GBEST} - X_i) \quad (1)$$

$$X_{i+1} = X_i + V_{i+1} \quad (2)$$

onde:

X = posição da partícula

V = velocidade

rnd() = número aleatório com distribuição uniforme no intervalo [0,1]

PBEST = melhor posição individual

GBEST = melhor posição social

i = intervalo de tempo atual

O coeficiente de valor 2 na fórmula original (1) foi utilizado para fazer com que o número aleatório gerado tivesse média 1.

O esquema básico de um algoritmo PSO é apresentado na figura 5.

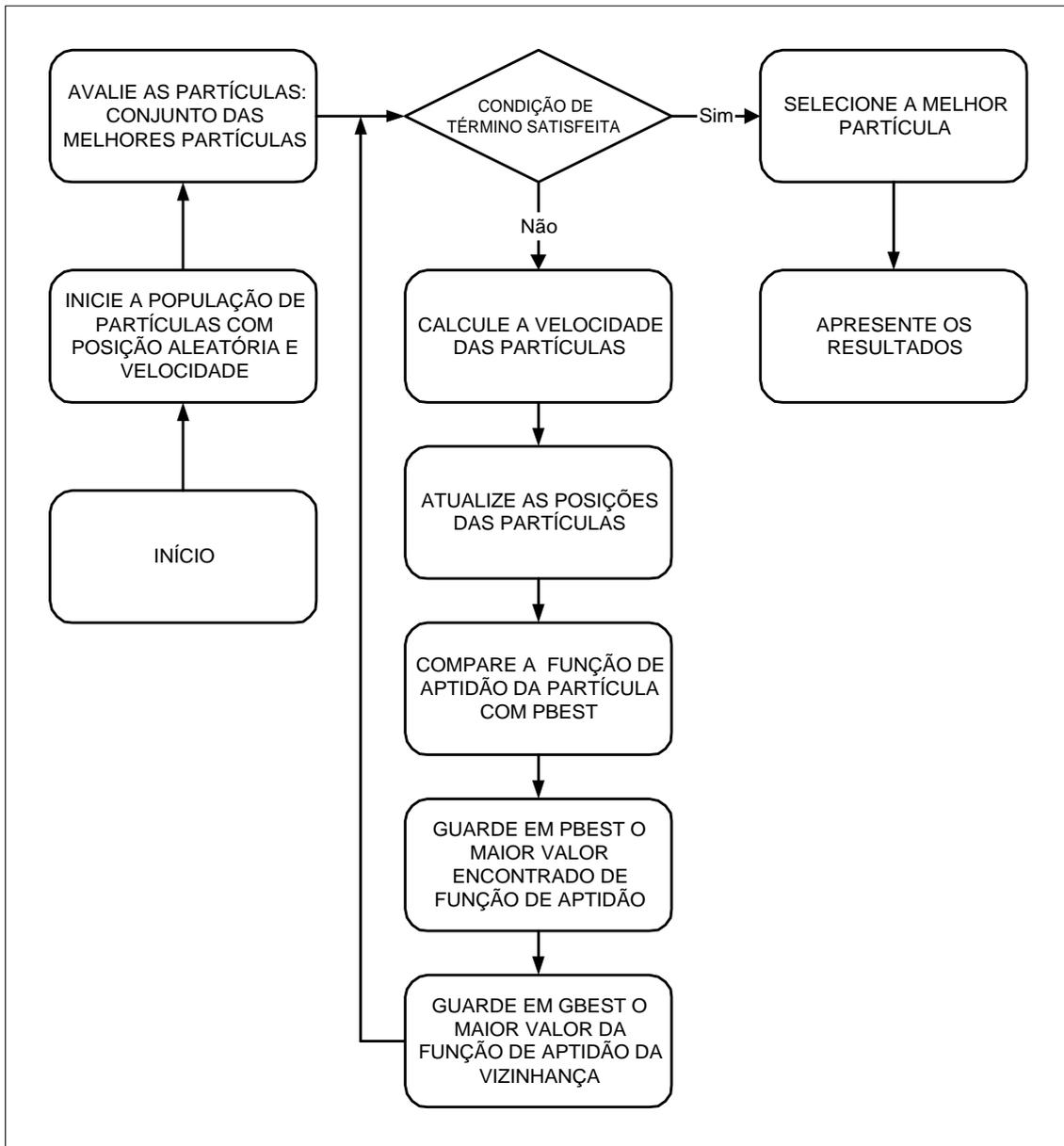


Figura 5 – Estrutura básica de um algoritmo PSO

O algoritmo PSO pode ser visto como um conjunto de vetores cujas trajetórias oscilam em torno de uma região definida por cada melhor posição individual e a melhor posição dos outros indivíduos [KENNEDY & EBERHART, 2001].

Para evitar que a velocidade cresça indefinidamente Eberhart e Kennedy [EBERHART & KENNEDY, 1995] sugerem que a velocidade máxima de cada partícula seja limitada ao valor limite de cada variável, ou seja:

$$V_{\text{máx}} = X_{\text{máx}} - X_{\text{min}} \quad (3)$$

Onde:

$X_{\text{máx}}$ – limite superior da variável livre

X_{min} – limite inferior da variável livre

A cada iteração a velocidade de cada partícula deverá ser limitada a este valor.

5.3.1 – Inércia

As simulações iniciais do algoritmo básico já indicavam a necessidade de um mecanismo de controle da velocidade da partícula e por isso foi necessário a imposição de um valor máximo $V_{\text{máx}}$. Se a velocidade excedia esse limite, à mesma era atribuída o valor de $V_{\text{máx}}$. Esse parâmetro provou ser crucial porque grandes valores de velocidade poderiam resultar em partículas passando de boas soluções enquanto valores pequenos poderiam resultar em uma exploração insuficiente do espaço de busca. Entretanto, a falta de mecanismo de controle da velocidade mais elaborado permitia uma baixa eficiência do PSO quando comparado com outras técnicas de computação evolucionária [ANGELINE, 1998].

Eberhart e Shi [EBERHART & SHI, 1998] introduziram no algoritmo básico um novo mecanismo de controle da velocidade com a incorporação de um parâmetro denominado peso inercial da partícula W . Os coeficiente fixos de valor 2 também foram alterados para $C1$ e $C2$, duas constantes de aceleração positivas chamadas parâmetro cognitivo e parâmetro social respectivamente. Com esses novos parâmetros a nova fórmula do cálculo da velocidade é dada por:

$$V_{i+1} = W * V_i + C1 * \text{rnd}() * (\text{PBEST} - X_i) + C2 * \text{rnd}() * (\text{GBEST} - X_i) \quad (4)$$

Onde:

W – coeficiente de peso inercial

C1 – parâmetro cognitivo

C2 – parâmetro social

O peso da inércia W é empregado para controlar o impacto do histórico prévio de velocidades no valor da velocidade corrente que é determinante no tipo de exploração executada pela partícula. Um alto valor de peso inercial facilita a exploração global (busca de novas áreas) enquanto um valor pequeno de peso inercial tende a facilitar a exploração local realizando uma busca detalhada em uma determinada área. A seleção de valores adequados do peso da inércia W pode fornecer um equilíbrio entre as necessidades de exploração global e local e dessa forma reduzir o número médio de iterações para localização do ótimo [EBERHART & SHI, 1998].

Mendes [MENDES, 2004] afirma que estudos recentes apontam para a fixação do valor do peso da inércia em torno de 0,5 como um bom valor para W .

Segundo Parsopoulos [PARSOPOULOS & VRAHATIS, 2002], os valores dos parâmetros, cognitivo $C1$ e parâmetro social $C2$, inicialmente propostos como $C1=C2=2$, já apresentaram resultados experimentais que indicaram que $C1=C2=0,5$ podem fornecer bons resultados. Outros estudos indicam a preferência pelo uso de valores do parâmetro cognitivo $C1$ maior que o parâmetro social $C2$, mas com $C1 + C2 \leq 4$ [CARLISLE & DOZIER, 2001].

5.3.2 – Variação Linear da Inércia

O desempenho com PSO na convergência para o ponto ótimo nas iterações finais tem na componente inercial da velocidade seu principal fator. Para tentar melhorar esse desempenho Eberhart [EBERHART & SHI, 2000] propôs que o valor de W fosse uma variação linear entre os valores 0,9 e 0,4 e o emprego destes valores durante as iterações demonstrou um aumento de desempenho do PSO. A variação linear proposta pode ser calculada por:

$$W_i = (W_{ini} - W_{fin}) * [(N - i)/N] + W_{fin} \quad (5)$$

Onde:

W_{ini} – Valor inicial do peso da inércia

W_{fin} – valor final do peso da inércia

N – número total de iterações

Algumas simulações utilizando o peso da inércia inicial em torno de 1,2 e declinando ao longo das iterações até o valor zero pode ser considerado uma boa escolha para W [PARSOPOULOS & VRAHATIS, 2002].

5.3.3 – Variação Não Linear da Inércia

Os conceitos relativos à exploração global e exploração local associados à variação linear do peso da inércia proposta por Eberhart [EBERHART e SHI, 2000] foram expandidos por outra proposta Chatterjee [CHATTERJEE & SIARRY, 2006] de variação não linear do peso da inércia.

Nessa nova proposta os valores do peso da inércia inicial (W_{ini}) e final (W_{fin}) permanecem inalterados, entretanto, ao longo das iterações o valor do peso inercial W pode ser mais bem controlado pela utilização de um expoente de não linearidade. A fórmula de cálculo proposta é;

$$W_i = [(N - i)^n / N^n] * (W_{ini} - W_{fin}) + W_{fin} \quad (6)$$

Onde:

W_i – valor inicial do peso da inércia para a iteração i

N – número total de iterações

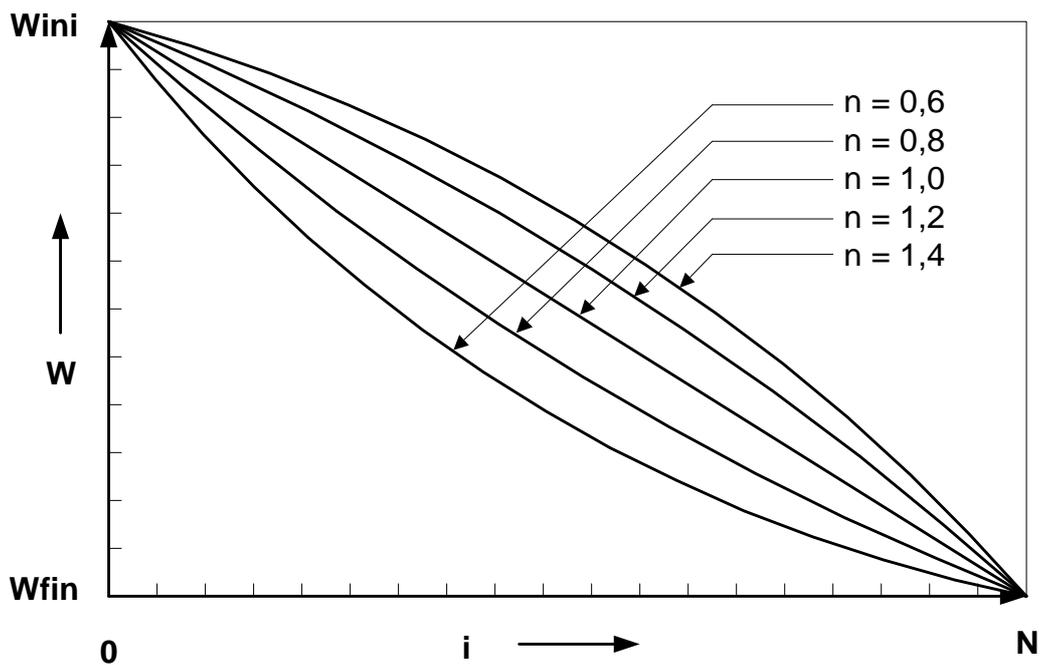
i – iteração atual

W_{ini} – valor inicial do peso da inércia

W_{fin} – valor final do peso da inércia

n – expoente de não linearidade

A figura 6 apresenta o comportamento da variação não linear do peso da inércia.



Fonte: [CHATTERJEE & SIARRY, 2006]

Figura 6 – Gráfico da variação não linear do peso da inércia

5.3.4 – Variação Fuzzy da Inércia

Outro tipo de controle do peso da inércia foi apresentado por Shi [SHI & EBERHART, 2001], onde foi utilizado um controlador fuzzy para geração do valor do

peso inercial. O controlador foi elaborado com nove regras, duas entradas e uma saída. Cada entrada e saída possui três conjuntos fuzzy definidos. Uma entrada era a função de aptidão da melhor posição social da geração corrente e a outra era o peso da inércia corrente. A saída era o novo peso da inércia.

Foi realizado um teste de desempenho utilizando a função de Rosenbrock com inicialização assimétrica e os resultados obtidos com o controlador fuzzy do peso da inércia mostraram um ganho de desempenho no PSO.

5.3.5 – Coefficiente de Constrição

Da mesma forma que o coeficiente do peso inercial proposto por Eberhart [EBERHART & SHI, 2000], Clerc [CLERC, 1999] também propôs um limitador da velocidade inércia chamado de Fator de Constrição. Este fator limita a velocidade em função dos coeficientes do parâmetro cognitivo (C1) e parâmetro social (C2) e é calculado da seguinte forma:

$$V_{i+1} = K * [V_i + C1 * \text{rnd}() * (\text{PBEST} - X_i) + C2 * \text{rnd}() * (\text{GBEST} - X_i)] \quad (7)$$

$$X_{i+1} = X_i + V_{i+1} \quad (8)$$

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (9)$$

Onde:

$$K \in [0,1]$$

$$\varphi = C1 + C2$$

$$\varphi > 4$$

Eberhart [EBERHART & SHI, 2000] fizeram um estudo comparativo de desempenho entre o PSO utilizando o coeficiente de peso inercial e com fator de restrição proposto por Clerc [CLERC, 1999]. Nas conclusões é mencionado que o algoritmo com fator de restrição pode ser considerado como um caso particular do algoritmo com peso inercial desde que os três parâmetros (W , K e φ) estejam conectados através da equação e das restrições impostas pelo fator de restrição. Dessa forma os resultados podem ser considerados semelhantes sempre que o algoritmo do fator de restrição for utilizado com $V_{\text{máx}} = X_{\text{máx}}$ e o algoritmo de peso inercial for utilizado com W , $C1$ e $C2$ obedecendo às restrições da equação do fator de restrição.

Os experimentos comparativos realizados em [EBERHART & SHI, 2000] concluem que o uso de $V_{\text{máx}}$ limitado ao valor de $X_{\text{máx}}$ (alcance dinâmico de cada variável em cada dimensão) apresentou resultados surpreendentes quando empregado em conjunto com o fator de restrição.

5.3.6 – Outras Variações na PSO

O algoritmo de enxames de partículas inicialmente proposto por Kennedy [KENNEDY & EBERHART, 1995] e modificado por Eberhart e Shi [EBERHART & SHI, 1998] com a introdução dos conceitos do peso inercial da partícula “ W ” e dos parâmetros cognitivo “ $C1$ ” e parâmetro social “ $C2$ ” no cálculo da velocidade é conhecido academicamente como algoritmo de enxame de partículas canônico.

A especificidade do algoritmo da PSO para os problemas de otimização com variáveis reais reduziu sua aplicabilidade em outras representações. A partir dessa necessidade diversos pesquisadores têm transformado o algoritmo dito canônico para operarem em espaços binários (Binary PSO), com variáveis discretas (Discrete PSO), com funções dinâmicas ou otimização multiobjetivo (MO-PSO). Essas abordagens fazem parte do largo campo de pesquisa criado inicialmente por Eberhart e Kennedy e fogem ao escopo do presente estudo.

5.3.7 – Os parâmetros do algoritmo da PSO

No algoritmo da PSO existem vários parâmetros explícitos cujos valores podem ser ajustados para produzir variações no modo como o algoritmo executa a busca no espaço do problema. A seguir os principais parâmetros da PSO serão revistos e dimensionados de acordo com recomendações acadêmicas.

a) Tamanho da população

O tamanho da população é dependente do problema. Tamanhos de população entre 20 e 50 partículas são os valores mais usuais. Desde cedo o algoritmo PSO demonstrou necessitar de pequenas populações em relação a outros algoritmos evolucionários para atingir uma alta qualidade na solução [MENDES, 2004].

b) Peso da inércia (W)

A regra do peso da inércia W na equação (4) é considerada crítica no comportamento da convergência do algoritmo da PSO. Eberhart [EBERHART & SHI, 2000] propôs que o valor de W fosse uma variação linear entre os valores 0,9 e 0,4.

c) Constantes de aceleração (C1 e C2)

O parâmetro cognitivo “C1” e o parâmetro social “C2”, também denominadas constantes de aceleração, não são críticos para a convergência da PSO, entretanto, quando bem dimensionados podem resultar em uma rápida busca e saída de mínimos locais. Geralmente C1 e C2 são utilizados com o mesmo valor, ou seja $C1 = C2 = \Phi/2$. Há um consenso entre os pesquisadores que para o valor de $\Phi = 4,1$, implicaria em $C1 = C2 = 2,05$.

d) Funções aleatórias (rnd())

As funções “rnd()” são usadas para preservar a diversidade da população e são uniformemente distribuídas no intervalo [0,1].

e) Coeficiente de constrição (K)

O coeficiente de constrição “K” controla a magnitude das velocidades e, de modo semelhante, o parâmetro $V_{\text{máx}}$ resulta em uma variante da PSO em relação ao algoritmo com utilização do peso inercial. A partir do consenso dos pesquisadores no valor de $\Phi=4,1$, implica no valor de $K = 0,729$

f) Velocidade máxima ($V_{\text{máx}}$)

O controle da velocidade é um assunto bastante complexo entre os pesquisadores, principalmente a partir do uso do coeficiente de constrição, entretanto, aqueles que utilizam $V_{\text{máx}}$ sempre o limitam à dimensão do alcance dinâmico de cada variável em cada dimensão [EBERHART & SHI, 2000].

g) Tamanho da vizinhança

O tamanho da vizinhança em um enxame é o número de partículas vizinhas que influenciam o movimento da própria partícula. Alguns estudos indicam que uma pequena vizinhança local é melhor para evitar mínimos locais e que uma vizinhança global induz a uma rápida convergência, entretanto esse parâmetro e a topologia de ligação entre as partículas ainda tem seu comportamento pouco detalhado pelos pesquisadores.

5.4 - SOCIOMETRIA DE ENXAME DE PARTÍCULAS

A vizinhança de uma partícula compreende um subconjunto de outras partículas do enxame com as quais a partícula possui comunicação direta. Essa sociometria do enxame é objeto de contínua pesquisa por [KENNEDY, 1999], [KENNEDY, 2000], [KENNEDY & MENDES, 2002] e [RICHARDS & VENTURA, 2004].

No estudo do enxame de partículas o termo topologia refere-se à representação gráfica da sociometria do enxame, ou seja, das inter-relações entre as partículas e as relações individuais de cada partícula com as demais partículas.

A pesquisa de enxame de partículas tem se socorrido de várias estruturas sociais simples, em particular a interação de indivíduos com seus vizinhos imediatamente adjacentes e a interação de todos os indivíduos com o melhor desempenho individual da população, embora outras estruturas sociais sejam possíveis. O desempenho de uma partícula isolada já foi mostrado que é insuficiente e que a interação entre as partículas é que faz o algoritmo trabalhar [KENNEDY & EBERHART, 2001].

Segundo Watts [WATTS, 1999], [WATTS & STROGATZ, 1998] o fluxo de informações em redes sociais é afetado por vários aspectos característicos de redes. Destacam-se as seguintes características:

a) Grau de conectividade

Essa medida representa a quantidade de conexões existentes entre os nós da rede. Cada partícula no enxame de partículas identifica individualmente o melhor ponto encontrado entre os seus k vizinhos. Dessa forma a variável k distingue as topologias LBEST da GBEST, muito provavelmente afeta o desempenho do fluxo de informações da rede.

b) Quantidade de clusters

O fator quantidade de clusters, denominado de C por Watts ocorre quando um vizinho de um nó também é vizinho de outro nó. O número de vizinhos em comum pode ser contado por nó e seu valor médio pode ser avaliado no grafo.

c) Distância mínima entre nós

A média da distância mínima entre os nós é uma importante característica para determinação da propagação da informação através da rede. Essa medida é altamente correlacionada com k e C .

5.4.1 – Topologias Estáticas

As partículas têm sido estudadas historicamente em dois tipos principais de topologias de vizinhança, chamadas GBEST e LBEST, já apresentadas na figura 4.

a) Topologias padrão

Na vizinhança GBEST, também denominada topologia Estrela, cada partícula individual é atraída pela melhor solução achada por qualquer membro do enxame. A estrutura dessa configuração é equivalente a uma rede social totalmente conectada; cada partícula é capaz de comparar o desempenho de todas as demais partículas do enxame, imitando a melhor.

Na vizinhança LBEST, também denominada topologia Círculo, cada partícula individual é afetada pelo melhor desempenho de seus k imediatos vizinhos na topologia da vizinhança. Em um caso LBEST comum, $k = 2$, a partícula só é afetada somente pelos seus vizinhos imediatamente adjacentes.

Na topologia Círculo, conjuntos de partículas estão distantes de outros conjuntos de partículas e os conjuntos também são independentes, mas os vizinhos próximos são conectados. Então, um segmento da população pode convergir para um mínimo local enquanto outro segmento converge para um ótimo diferente ou permanece procurando. Nesta topologia a informação é propagada de vizinho para vizinho e se um ponto ótimo é realmente o melhor encontrado por qualquer parte da população ele eventualmente atrairá todas as demais partículas.

b) Outras topologias

Além das topologias clássicas GBEST e LBEST, outras formas de topologias têm sido propostas como Von Neumann, Roda, Pirâmides e Formas randômicas.

A arquitetura Von Neumann possui a forma de uma grade e considera que a partícula possui vizinhos imediatos acima, abaixo, à esquerda e à direita, ou seja, $K = 4$. O comportamento dessa topologia em termos de propagação da informação é superior ao da topologia LBEST.

Na topologia em Roda as partículas são efetivamente isoladas individualmente uma das outras. Toda a informação é comunicada por meio de uma partícula central ou focal que compara o desempenho de todas as partículas da população e ajusta sua

trajetória em direção da melhor partícula. Se esse ajuste resulta em um melhor desempenho da partícula focal então o desempenho é eventualmente comunicado ao resto da população. Dessa forma a partícula focal serve como um tipo de buffer ou filtro reduzindo a velocidade de transmissão das boas soluções para toda a população. O efeito de buffer da partícula focal previne a rápida convergência para um mínimo local e deste modo preserva a diversidade de potenciais soluções do problema [KENNEDY & EBERHART, 2001].

A topologia em forma de pirâmides apresenta a forma tri-dimensional com clusters de nós esparsamente dispostos e as topologias randômicas são criadas aleatoriamente baseadas em conceitos de grafos heterogêneos. Essas topologias podem ser consideradas como experiências isoladas de alguns pesquisadores.

A figura 7 apresenta a representação gráfica das topologias Von Neumann e Roda.

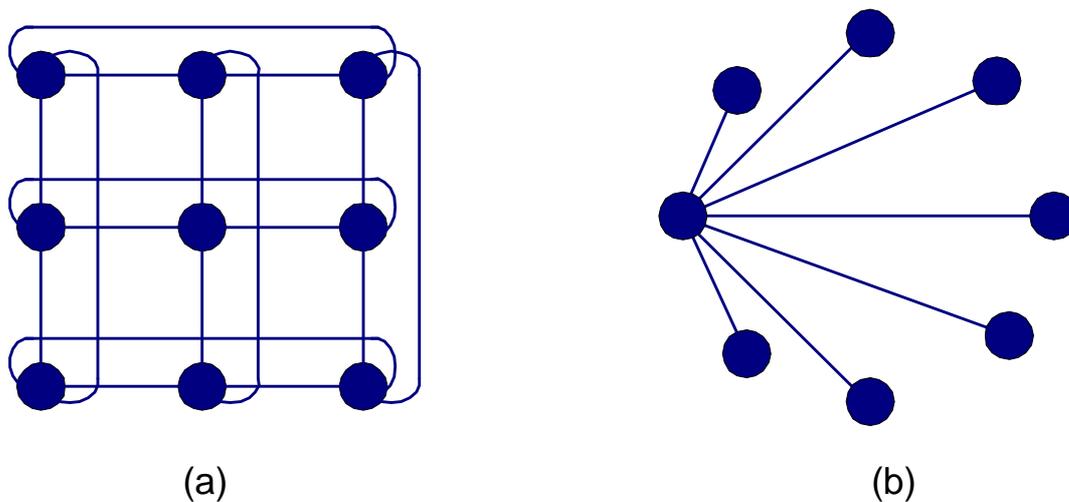


Figura 7 – Representação gráfica das topologias (a) Von Neumann e (b) Roda

c) Resultados observados

Desde as primeiras utilizações das topologias ditas padrão, GBEST e LBEST, os resultados dos experimentos têm demonstrado que na topologia GBEST há uma tendência de rápida convergência com uma inconveniente característica de atração por

mínimos locais, já que a trajetória de cada partícula individual é diretamente influenciada pelo melhor ponto encontrado por qualquer membro da população [KENNEDY & MENDES, 2002].

Na topologia LBEST o fluxo da informação é extremamente reduzido em relação à GBEST. Esta topologia tem a vantagem de permitir uma busca paralela, com subpopulações podendo convergir para diversas regiões no espaço de busca, onde, equitativamente, pontos ótimos podem estar localizados. Isso permite uma melhor distribuição das partículas nas boas regiões. Entretanto, se uma região é melhor que a outra ela irá atrair as demais partículas. Dessa forma a busca paralela é uma estratégia de busca com convergência mais lenta que a topologia GBEST, porém menos vulnerável a mínimos locais [POLI, KENNEDY, BLACKWELL, 2007].

Diversos pesquisadores têm procurado solução para o problema da convergência prematura do algoritmo padrão da PSO por meio da pesquisa de novas topologias. Entretanto, as observações contidas nas conclusões desses estudos indicam que há muito trabalho a ser realizado até que os resultados possam ser mais claramente compreendidos.

Investigações realizadas no paradigma do enxame de partículas têm concluído que o efeito da topologia da população interage com a função que está sendo otimizada. Alguns tipos de população trabalham bem em algumas funções; outras trabalham melhor em outras funções [KENNEDY, 1999].

Em relação ao trabalho de Kennedy [KENNEDY, 1999] comparando diversas topologias (círculo, roda estrela, etc) em um conjunto de funções padrão, Poli [POLI, KENNEDY, BLACKWELL, 2007] ressalta que a mais importante conclusão foi a de que realmente existem diferenças de desempenho de acordo com a topologia implementada. Essas diferenças dependem da função testada, e não houve uma conclusão que sugira qual das topologias foi efetivamente melhor que a outra.

No trabalho de Kennedy e Mendes [KENNEDY & MENDES, 2002] numerosos aspectos da topologia de redes sociais foram testados na avaliação de desempenho do algoritmo de enxame de partículas. Por exemplo: a verificação da influência da própria

partícula na sua vizinhança; a geração de 1343 grafos baseados em critérios de grau de conectividade e quantidade de clusters [WATTS, 1999] e os desvios-padrão dessas duas medidas para a geração de estruturas homogêneas e heterogêneas. Nas conclusões desse trabalho destacam-se:

- A recomendação de utilização da estrutura da topologia Von Neumann pela consistência do desempenho observado. Esta topologia possui o mesmo tipo de paralelismo da topologia LBEST, contudo com grau de conectividade $K = 4$; dessa forma a rede é mais densamente conectada que LBEST, mas menos conectada que em GBEST. Esse resultado já era esperado se efetivamente a conectividade é mínima em LBEST e máxima em GBEST.

- Apesar de a pesquisa realizada ter identificado configurações de populações superiores, não foi possível nomear os fatores topológicos que resultaram em um melhor desempenho de uma determinada topologia para um determinado conjunto de funções.

A mais completa exploração dos efeitos da topologia nos algoritmos de enxame de partículas foi a tese de doutorado de Mendes [MENDES, 2004]. Mendes utilizou um novo método de interação entre as partículas, FIPS (Fully Informed Particle Swarm) onde cada partícula é influenciada por uma média estocástica das melhores posições prévias de todos os seus vizinhos ao invés da melhor solução entre todos os vizinhos e comparou esses resultados com o algoritmo canônico da PSO em diversos tipos de topologias. Nesse estudo foi utilizado um processo de padronização (standardização) de medidas de desempenho utilizando uma função de média 0 e desvio-padrão 1. Outra novidade foi a utilização de duas medidas de desempenho; a primeira, definida como uma medida de busca local caracterizada pela habilidade do algoritmo atingir um determinado gradiente; e a segunda, uma medida de habilidade de busca global definida com a proporção de sucesso que um critério foi atingido. Sobre as conclusões dessa tese destacam-se as observações de [POLI, KENNEDY, BLACKWELL, 2007].

- A evidência de que a rede social e os efeitos da topologia dependem do modo de interação entre as partículas.

- Surpreendentemente não houve evidência de correlação entre as medidas de desempenho utilizadas na análise das topologias utilizadas versus modo de interação entre as partículas (tipo canônico e tipo FIPS). Isto corresponde à afirmação que a habilidade de busca local não corresponde, para uma mesma configuração, à habilidade de busca global. As duas medidas de desempenho, em geral, são independentes de onde se pode concluir que a topologia que produz uma boa busca local foi muito diferente daquela que produziu uma boa habilidade de busca do ótimo global.

No algoritmo canônico da PSO as topologias são estáticas. Entretanto, os resultados encontrados nas topologias estáticas sugerem que o uso de topologias adaptativas ou dinâmicas pode ser uma solução e vários pesquisadores estão trabalhando nessa área.

5.4.2 – Topologias Dinâmicas

Com o avanço das pesquisas das topologias de vizinhança de algoritmos de enxame de partículas, é cada vez mais difundido o conceito da não existência de uma topologia predominante hábil para execução de busca global e busca local em algoritmos de enxame de partículas. Ainda que alguns resultados estatísticos tenham indicado que algumas topologias possam ser superiores em algumas características, não há definição de quais fatores permitiram um melhor desempenho topológico nos experimentos.

Nos últimos anos têm surgido diversos estudos relativos à flexibilização da topologia ao longo das corridas, dentre os quais se destacam:

a) Operador de Vizinhança

Suganthan [SUGANTHAN, 1999] propõe a introdução de uma variável denominada operador de vizinhança. Durante os estágios iniciais da otimização a vizinhança no algoritmo da PSO deve ser a própria partícula. Com o aumento do número de gerações, a vizinhança deverá ser gradualmente estendida para incluir todas as partículas. Em outras palavras, a variável GBEST no algoritmo da PSO é trocada por

LBEST (melhor solução local) onde o tamanho da vizinhança local é gradualmente aumentado.

O operador de vizinhança é calculado a cada atualização de partículas e utiliza uma relação entre a distância de todas as partículas da população (partículas candidatas) até a partícula em atualização e distância máxima entre as partículas candidatas. Essa relação é comparada com uma ponderação da relação entre o número da iteração e total de iterações para a decisão de utilização de LBEST com a introdução da partícula candidata ou de GBEST.

Além de o operador de vizinhança atuar a cada cálculo da vizinhança da partícula, o estudo menciona também a atualização por decaimento dos valores do coeficiente de peso inercial (W), e parâmetro cognitivo ($C1$) e parâmetro social ($C2$). Nesse caso é utilizado um fator de decaimento traduzido pela razão entre a iteração corrente (número da geração) e o total de iterações (total de gerações).

b) Razão de Distância Aptidão

Peram (PERAM et al., 2003) usou um método baseado na distância Euclidiana e no valor da função de aptidão para o cálculo da influência da iteração com uma partícula vizinha.

Para cada partícula, o método calcula qual partícula possui o mais alto valor de FDR (“Fitness Distance Ratio”), ou seja, a razão da diferença entre o valor da função de aptidão da partícula e a função de aptidão da partícula vizinha e a diferença de distância entre a partícula e a partícula vizinha na mesma dimensão do espaço de busca.

A partir do cálculo do maior valor de FDR é possível garantir que o vizinho selecionado possui um bom desempenho e uma reduzida probabilidade de que a partícula em atualização possa estar interagindo com uma partícula vizinha de uma região distante no espaço de busca. Não há garantia que o mesmo vizinho seja selecionado para cada dimensão.

No novo algoritmo a atualização da velocidade será influenciada por três fatores: a melhor experiência da partícula (PBEST), a melhor experiência global (GBEST) e a melhor experiência do vizinho mais próximo (BEST NEAREST).

c) Hierarquia Dinâmica

Janson e Middendorf [JANSON & MIDDENDORF, 2005] propõem uma versão hierárquica da PSO (H-PSO), onde uma partícula é influenciada pela sua melhor posição e pela melhor posição de uma partícula hierarquicamente superior.

Na H-PSO todas as partículas são arranjadas em uma árvore de forma hierárquica com cada nó da árvore contendo apenas uma partícula. Esse arranjo tem como objetivo dar as melhores partículas do enxame uma alta influência. De acordo com seus desempenhos as partículas se movem para cima e para baixo na hierarquia, isto é, se uma partícula está em um nó filho e é encontrada uma solução que é melhor que a solução da partícula do nó pai, então essas partículas são trocadas no nó pai.

Para aumentar o sucesso da busca é mencionado que a estrutura da hierarquia é dinamicamente alterada, gerando com isso algumas variantes da H-PSO com árvores de diferentes graus e árvores com graus variáveis ao longo da corrida.

d) TRIBES

Clerc, [CLERC, 2006] e [CLERC 2007] desenvolveu um parâmetro livre denominado TRIBES, no qual detalhes da topologia, incluindo o tamanho da população, evoluem ao longo do tempo em resposta do desempenho encontrado nas partículas. A população é dividida em subpopulações, cada uma mantendo sua disposição e estrutura. Essas subpopulações, denominadas tribes, quando classificadas como “good tribes” podem ser beneficiadas pela remoção de partículas com baixo desempenho, já que elas possuem boas soluções do problema podem permitir-se reduzir sua população. De outra forma as “bad tribes” podem ser beneficiadas com a adição de novas partículas, aumentando a possibilidade de melhoria de desempenho. As novas partículas são geradas aleatoriamente. Esse método reavalia e modifica a estrutura da população a cada $L/2$ iterações, onde L é o número de conexões da população.

e) Sociometria Dinâmica

Richards e Ventura [RICHARDS & VENTURA, 2003], buscando um equilíbrio entre os esforços de busca global e busca local no algoritmo da PSO, propuseram um aumento das conexões existentes na rede ao longo da corrida.

Nesse método o enxame de partículas é inicializado como uma topologia tipo Círculo, entretanto é mencionado que cada partícula é conectada a apenas outra partícula do enxame, ou seja, $k=1$. Ao longo do tempo, conexões adicionais são adicionadas. Conseqüentemente, a rede final estará totalmente conectada em uma topologia tipo Estrela. A estratégia de implementação adotada foi de o enxame estar totalmente conectado quando 4/5 do total de funções de avaliação tiver sido realizado. Antes desse tempo, uma conexão é adicionada a cada partícula a intervalos regulares. É mencionado que as conexões são realizadas de forma assimétrica.

f) Vizinhaça Complexa

Godoy e Von Zuben [GODOY & VON ZUBEN, 2009], com o objetivo de evitar a convergência prematura e aumentar a capacidade de exploração do algoritmo da PSO, propuseram uma variação denominada CNPSO (“Complex Neighborhood Based Particle Swarm Optimization”). O novo algoritmo apresenta um mecanismo de evolução topológica utilizando conceitos de redes complexas com conexões que variem no tempo para a definição da vizinhaça das partículas.

Inicialmente a vizinhaça das partículas é gerada por um modelo matemático de redes complexas denominado SF (“scale-free”) desenvolvido por Barabási e Albert [BARABÁSI & ALBERT, 1999] onde o número de arestas ligadas a um nó (grau do nó) é proporcional à sua importância ou desempenho na rede. O modelo é gerado a partir das informações do tamanho da rede (N), ou seja, o número total de nós e o número máximo de arestas (m).

Pela forma com que os graus dos nós se distribuem, a rede complexa resultante acaba por ser altamente dependente dos hubs da rede (nós com maior número de conexões). Portanto, caso um desses nós localize-se por muito tempo em uma região

com baixa qualidade, o desempenho de todo o algoritmo será degradado. Para isso foi estabelecido um modelo de evolução topológica capaz de penalizar partículas com pior histórico na busca, através da perda gradual de conexões. Desta forma, após uma quantidade determinada de iterações da rede, o sistema seleciona uma aresta para sofrer reconexão, sendo desligada de sua extremidade com pior desempenho e conectada a um nó selecionado aleatoriamente. Ao selecionar a nova extremidade da aresta usando um critério puramente aleatório, busca-se evitar que apenas um nó retenha parte significativa das arestas redirecionadas, o que tende a ocorrer quando tal nó é escolhido com base em seu desempenho de busca.

5.5 - VARIANTES DA PSO

Nesse item serão apresentadas algumas variantes do algoritmo da PSO. Algumas dessas variantes têm sido propostas para incorporar capacidades de outras técnicas de computação evolucionária, tais como versões híbridas da PSO (PSO Híbrido) ou adaptação de parâmetros da PSO (PSO Adaptativo). Em outros casos a natureza do problema a ser resolvido requer que a PSO trabalhe sob condições de ambientes complexos (problemas multiobjetivo, problemas de otimização com restrições ou sistemas de acompanhamento dinâmico). Também serão apresentadas variantes discretas da PSO e outras variações do algoritmo original que foram incluídas para aumentar o desempenho (PSO dissipativo) e técnicas de alongamento, congregação passiva, cooperação e aprendizagem compreensiva para prevenir a armadilha dos mínimos locais, Del Valle (DEL VALLE et al., 2008).

5.5.1 – PSO Híbrido

O PSO híbrido incorpora métodos já testados em outras técnicas de computação evolucionária. O principal objetivo desse tipo de PSO é aumentar a diversificação da população. Como exemplo podemos citar: a inclusão dos algoritmos genéticos na PSO (GA-PSO), por El-Dib (EL-DIB et al., 2004) e Naka (NAKA et al. 2003), PSO evolucionária (EPSO) por Angeline (ANGELINE, 1998), PSO com evolução diferencial (DEPSO) por Zhang e Xie (ZANG & XIE, 2003) e PSO composto (C-PSO) por Kannan (KANNAN et al., 2004), etc.

5.5.2 – PSO Adaptativo

São inúmeros PSO baseados em ajustes de parâmetros do algoritmo original, como aleatoriedade no componente do peso de inércia Eberhart e Shi (EBERHART & SHI, 2001), ajuste no número de partícula e tamanho da vizinhança por Zhang (ZHANG et al., 2003), etc.

5.5.3 – PSO em Ambientes Complexos

São PSO com necessidades específicas como por exemplo: problemas de otimização multiobjetivo por Parsopoulos e Vrahatis (PARSOPOULOS & VRAHATIS, 2002), problemas com manipulação de restrições por Hu (HU, 2006) e problemas de acompanhamento dinâmico por Hu e Eberhart (HU & EBERHART, 2002), etc.

5.5.4 – PSO Discreto

São PSO baseadas em modificações da versão binária da PSO para aumentar a performance do algoritmo em aplicações de normalização, decisão, decodificação destacando-se SHi (SHI, 2004).

5.5.5 – Outras variantes da PSO

a) PSO Gaussiana (GPSO)

São algoritmos onde funções gaussianas são utilizadas para guiar o movimento das partículas destacando-se Secret e Lamont (SECRET & LAMONT, 2003).

b) PSO Dissipativo (DPSO)

São algoritmos onde estruturas dissipativas são utilizadas para solução de entropias negativas introduzidas propositalmente no cálculo de velocidades e posições das partículas destacando-se Xie (XIE et al., 2002).

c) PSO com Congregação Passiva (PSOPC)

São algoritmos que utilizam um mecanismo de agregação em grupos de animais e incorporam ao algoritmo original um novo termo na equação de atualização da velocidade, destacando-se: He (HE et al, 2004), Vlachogiannis e Kwang (VLACHOGIANNIS & KWANG, 2006) e Zhihua (ZHIHUA et al., 2009).

d) PSO com Alongamento (SPSO)

Nesse tipo de algoritmo são utilizados dois estágios de transformação da função de aptidão, eliminando no primeiro estágio todos os pontos mínimos encontrados para que no segundo estágio possa ser localizado o ótimo global, destacando-se Parsopoulos e Vrahatis (PARSOPOULOS & VRAHATIS, 2002) e Kannan (KANNAN et al, 2004).

e) PSO Cooperativo (CPSO)

Nesse algoritmo, múltiplos enxames são utilizados em cooperação na otimização de uma única função, destacando-se Bergh e Engelbrecht (BERGH & ENGELBRECHT, 2004).

f) PSO com Aprendizagem Compreensiva (CLPSO)

Esse algoritmo propõe que ao longo das iterações cada partícula possa em cada dimensão ser “guiada” por diferentes partículas selecionadas aleatoriamente, destaca-se Liang (LIANG et al., 2006)

6 - PROPOSTA

No presente trabalho uma nova variante de PSO baseada em congregação passiva seletiva é proposta com o objetivo de aumentar o desempenho da PSO padrão. O mecanismo de comportamento biológico de grupos de congregação passiva utiliza um processo seletivo coerente com a necessidade de melhoria contínua das iterações do enxame e permite a diversificação da transferência de informações entre os indivíduos do enxame. A simplicidade e o baixo uso de recursos computacionais, característicos das origens da PSO, foram mantidos no novo modelo.

6.1 - CONTEXTO DA PROPOSTA

Desde a introdução dos conceitos do enxame de partículas, o conhecimento técnico sobre o comportamento do algoritmo canônico ou padrão e suas principais melhorias vem sendo investigado pelos pesquisadores, principalmente em relação aos parâmetros do algoritmo da PSO detalhados no item 5.3.7, ou seja: tamanho da população, peso da inércia, constantes da aceleração, coeficiente de constrição, velocidade máxima, tamanho da vizinhança.

Há um conhecimento tácito acumulado da alta correlação entre o desempenho do algoritmo e a interação entre a partícula e sua vizinhança. O número de vizinhos ou tamanho da vizinhança afeta a velocidade de convergência do algoritmo e é geralmente aceito que um grande tamanho de vizinhança faz com que o algoritmo convirja mais rápido para a solução do problema, mas com a fraqueza da atração por mínimos locais, enquanto uma pequena vizinhança previne a partícula de uma convergência prematura, permitindo uma melhor exploração de diferentes regiões do espaço de busca.

Os resultados apresentados no item 5.4.1.c para as topologias estáticas concluíram que a habilidade de busca local não corresponde, para uma mesma configuração, à habilidade de busca global. Esse fato abriu o caminho para o estudo de topologias dinâmicas, também já apresentadas no item 5.4.2.

Nas pesquisas relativas às topologias dinâmicas, bem como em algumas variações da PSO (PSO híbrido, binário, dinâmico e variantes da PSO), observou-se uma tendência dos pesquisadores à utilização conjunta de novos modelos de topologia com algum tipo de mecanismo de geração de aleatoriedade nas partículas utilizadas como referência na vizinhança como forma de aumento da diversidade da população.

A aleatoriedade mencionada pode ser observada na flexibilização introduzida pelo operador de vizinhança proposto por Suganthan [SUGANTHAN, 1999], pela introdução do quarto termo da razão de distância aptidão proposta por Peram (PERAM et al., 2003), pela alteração dinâmica da hierarquia do enxame proposta por Janson e Middendorf [JANSON & MIDDENDORF, 2005], pela geração aleatória de partículas nas subpopulações proposta por Clerc [CLERC, 2006], pela introdução da partícula aleatória proposta por He (HE et al, 2004), bem como no modelo de evolução topológica proposto por Godoy e Von Zuben [GODOY & VON ZUBEN, 2009].

A necessidade de aumento da diversidade foi a forma encontrada pelos pesquisadores para combater a convergência prematura, notadamente crítica nas aplicações da PSO, na busca pelo equilíbrio das capacidades de busca global (“exploration”) e de busca local (“exploitations”). Entretanto, determinadas soluções trouxeram complexidade ao algoritmo padrão bem como aumento do esforço computacional.

6.2 - CONGREGAÇÃO PASSIVA SELETIVA

O algoritmo da PSO é inspirado em comportamento social de organismos tais como bando de pássaros, cardume de peixes, ou enxame de insetos. Há uma característica comum em relação à integridade espaço-temporal desses grupos de organismos: o grupo move-se como um todo sem perda da forma e densidade.

Parrish e Hamner (PARRISH & HAMNER, 1997) estudaram as forças biológicas de preservação da integridade dos grupos e formularam modelos matemáticos da estrutura espacial de grupos de animais para mostrar como os animais se organizam entre si. Segundo esses modelos, existem dois tipos principais de forças de

reunião de grupos de organismos: a reunião por forças físicas, externas e não sociais e a reunião por forças sociais.

A reunião de organismos por forças físicas é denominada agregação. Existem dois tipos de agregação: agregação passiva e agregação ativa.

Agregação passiva é a reunião de organismos por processos físicos. Um exemplo de agregação passiva é a agregação de plâncton em determinadas regiões do mar resultado da ação das forças físicas das correntes marítimas.

Agregação ativa é a reunião de organismos por um recurso atrativo, tais como comida, melhores condições climáticas para procriação ou espaço, com cada membro do grupo posicionado em um local específico

A reunião de organismos por processos sociais é denominada congregação. Existem dois tipos de congregação: congregação passiva e congregação ativa.

Congregação passiva é atração de um indivíduo por outros membros do grupo. Entretanto, observa-se que nesse tipo de reunião não há a apresentação de um comportamento social para que a atração aconteça. A congregação passiva é observada entre indivíduos com baixa fidelidade com o grupo porque a congregação pode ser composta de indivíduos com uma pequena ou nenhuma relação genética entre os mesmos. Isso é observado em cardumes de peixes, bando de pássaros. Na verdade, cada indivíduo apropria-se da vantagem da existência do grupo ou da vida em grupo, independente do destino dos seus vizinhos. Nesse tipo de reunião de organismos as informações podem ser transferidas, mais passivamente do que ativamente.

A congregação social geralmente está presente em grupos. Os membros possuem alta correlação entre si, divisão do trabalho por indivíduos especializados (colônias de insetos). Por exemplo, formigas usam contatos por antenas para transferir informações sobre identidade de indivíduos ou sua localização. Em colônias de formigas, abelhas, tarefas são executadas coletivamente por grupos de indivíduos especializados, que são mais eficientes na execução que indivíduos não especializados. A congregação social geralmente acontece quando a fidelidade ao grupo é alta.

He (HE et al, 2004) em sua utilização do mecanismo da congregação passiva menciona que o terceiro termo da equação de atualização da velocidade do algoritmo padrão da PSO, $C2 * \text{rnd}() * (\text{GBEST} - X_i)$, deve ser considerado como agregação ativa, pois GBEST representa a melhor solução do enxame encontrada até o momento (local com mais “comida”). Na sua proposta, um quarto termo $C3 * \text{rnd}() * (R_i - X_i)$ é introduzido, onde R_i é uma partícula selecionada aleatoriamente do enxame e C3 um coeficiente de congregação passiva. Esse novo termo cumpriria a função de transferência de informações empregando um modelo de congregação passiva.

Em alguns estudos, como o de Razão de Distância Aptidão (FDR) de Peram (PERAM et al., 2003), foi também introduzido um quarto termo na equação do algoritmo original da PSO. Sem a conceituação de congregação passiva, a nova partícula denominada (BEST NEAREST) é escolhida em função da razão da diferença entre o valor da função de aptidão da partícula e a função de aptidão da partícula vizinha e diferença de distância entre a partícula e a partícula vizinha na mesma dimensão do espaço de busca. Entretanto, em cada dimensão é escolhida uma partícula e não há garantias que em todas as dimensões a mesma partícula vizinha seja escolhida.

O presente trabalho inspira-se na necessidade observada em pesquisas anteriores da introdução de mecanismo de geração de aleatoriedade como uma forma de aumento da diversidade. A utilização de congregação passiva pode cumprir essa necessidade, entretanto, a forma de escolha da partícula que fornecerá maiores informações sobre o enxame não deve estar em desacordo com o princípio de melhoria contínua observada no algoritmo original da PSO.

Nos conceitos iniciais da PSO, cada partícula é atraída não somente pela melhor posição individual PBEST, como pela melhor posição social GBEST. A cada iteração, independente de onde se encontre a partícula na iteração anterior são as posições de PBEST e GBEST que conduzirão para nova posição da partícula. A melhoria contínua é a essência do processo de otimização por enxame de partículas, e por essa razão o modelo de congregação passiva seletiva proposto altera a aleatoriedade na escolha da partícula de congregação passiva, por entender que o uso de partículas com pior desempenho que o valor individual da própria partícula introduz um retrocesso na melhoria da posição da partícula objeto da movimentação.

A seleção da partícula de congregação passiva será realizada pela partícula cujo valor de aptidão seja imediatamente superior ao valor da aptidão da partícula individual. Dessa forma, impede-se a utilização de partículas com pior desempenho no processo, aliando-se também uma possível probabilidade de proximidade da partícula de congregação passiva.

$$\begin{aligned} V_{i+1} = W * V_i + C1 * \text{rnd}() * (\text{PBEST} - X_i) + \\ C2 * \text{rnd}() * (\text{GBEST} - X_i) \\ C3 * \text{rnd}() * (\text{SBEST} - X_i) \end{aligned} \quad (10)$$

Onde:

PBEST – melhor posição individual

GBEST – melhor posição social

SBEST – posição imediatamente superior a PBEST

6.2.1 - Pseudo-código da PSO com Congregação Passiva Seletiva

Para $e = 1$ até o número máximo de execuções (E_{max})

- Inicialização das posições e velocidades das partículas em todas as dimensões.
- Inicializa a melhor função de aptidão do enxame (P_g).

Para $r = 1$ até o número máximo de gerações (R_{max})

Se $r = 1$ (Primeiras avaliações das partículas)

- Calcula a função de aptidão inicial das partículas (P_i).
- Calcula a melhor posição da vizinhança (P_g).
- Armazena dados da melhor partícula (P_{best}).

Fim-se

Para $s = 1$ até o número máximo de partículas (S_{max})

- Calcula a posição da partícula de congregação social seletiva (P_{cps}).

Para $d = 1$ até D_{max}

- Calcule a atualização da velocidade da partícula (V_{id}).
- Limite a magnitude da velocidade da partícula.
- Atualize a posição da partícula (X_i).
- Limite a posição da partícula no espaço de busca.

Fim para d (Término das dimensões)

- Calcule a nova função de aptidão da partícula
- Se necessário atualiza melhor posição da partícula (P_i).
- Se necessário atualiza melhor posição da vizinhança (P_g).
- Armazena dados da melhor partícula (P_{best}).

Fim para s (Término das partículas)

- Armazena resultados da execução/geração.
- Armazena resultados se P_g atingiu as exigências do problema.

Fim para r (Término das gerações)

Armazena resultados da execução

Fim para e (Término das execuções)

Calcule índices estatísticos das execuções

6.3 - ESQUEMA EXECUÇÃO-GERAÇÃO-PARTÍCULA-DIMENSÃO

No presente estudo, optou-se pela definição dos termos de controle do algoritmo elaborado visando à padronização da linguagem, tanto na compreensão da metodologia de cálculo quanto na comparação de resultados com outras pesquisas. São as seguintes as definições utilizadas:

- a) Um teste é composto de E_{max} execuções ou corridas. Cada execução ou corrida é composta de R_{max} gerações ou iterações. Cada geração ou iteração é executada sobre S_{max} partículas e cada partícula é definida para um espaço n -dimensional de D_{max} dimensões.
- b) A dimensionalidade é definida como o número de variáveis de decisão em um problema de otimização.
- c) Uma geração ou iteração é a movimentação no espaço de busca de todas as partículas do enxame.
- d) A inicialização do enxame consiste no posicionamento e na definição de valor de velocidade aleatórios de cada partícula no espaço de busca do problema.
- e) Uma falha existe se, ao final de uma execução com R_{max} gerações, a diferença entre o valor mínimo da função e o valor mínimo encontrado na execução for superior ao erro máximo admissível da função. A taxa de falhas é calculada sobre o percentual de falhas ocorridas durante um teste com E_{max} execuções.

A tabela 1 apresenta o esquema da organização do programa para os controles de Execução-Geração- Partícula-Dimensão.

Execução e = 1 até E_max						
Inicialização			Geração r = 1 até R_max			
			Partícula	Dimensões		Melhores valores
			s = 1 até S_max	d = 1 até D_max		
X[1][d]	V[1][d]	P[1]	s = 1	novos V[1][d]	best P[1] Pbest	
				novos X[1][d]		
				novos P[1] ?		
				novos Pbest ?		
X[2][d]	V[2][d]	P[2]	s = 2	novos V[2][d]	best P[2] Pbest	
				novos X[2][d]		
				novos P[2] ?		
				novos Pbest ?		
X[3][d]	V[3][d]	P[3]	s = 3	novos V[3][d]	best P[3] Pbest	
				novos X[3][d]		
				novos P[3] ?		
				novos Pbest ?		
•	•	•	•	•	•	•
•	•	•	•	•	•	•
X[s][d]	V[s][d]	P[s]	s = S_max	novos V[s][d]	best P[s] Pbest	
				novos X[s][d]		
				novos P[s] ?		
				novos Pbest ?		

best	Rbest [e] [r]	Media_Rbest	Media_Ebest
Pbest	Ebest [e]	Melhor_Rbest	Taxa de sucesso

Tabela 1 – Esquema Execução-Geração- Partícula-Dimensão

6.4 – AVALIAÇÃO DO ALGORITMO

Para avaliar o desempenho do algoritmo proposto para Otimização por enxame de partículas com congregação passiva seletiva, o presente trabalho irá comparar os seguintes tipos de PSO:

- a) PSO padrão denominado PSO-P.
- b) PSO Clerc [CLERC, 1999] denominado PSO-C.
- c) PSO com congregação passiva de He (HE et al, 2004) denominado PSO-CP.
- d) PSO com congregação passiva seletiva denominado PSO-CPS.

Todos os tipos de PSO avaliados utilizaram a vizinhança global para o cálculo da melhor partícula de desempenho social (GBEST).

6.4.1 – Parâmetros Internos dos tipos de PSO

Serão utilizados os seguintes parâmetros de controle dos tipos de PSO avaliados:

- a) PSO padrão (PSO-P)

W0 – coeficiente de peso inercial inicial = 0,9

W1 – coeficiente de peso inercial inicial = 0,4

C1 – parâmetro cognitivo = 2,0

C2 – parâmetro social = 2,0

- b) PSO Clerc (PSO-C)

K – coeficiente ou fator de constrição = 0,729

C1 – parâmetro cognitivo = 2,05

C2 – parâmetro social = 2,05

- c) PSO com congregação passiva (PSO-CP)

W0 – coeficiente de peso inercial inicial = 0,9

W1 – coeficiente de peso inercial inicial = 0,7

C1 – parâmetro cognitivo = 0,5

C2 – parâmetro social = 0,5

C3 – parâmetro de congregação passiva = 0,5

d) PSO com congregação passiva seletiva (PSO-CPS)

W0 – coeficiente de peso inercial inicial = 0,9

W1 – coeficiente de peso inercial inicial = 0,4

C1 – parâmetro cognitivo = 1,9

C2 – parâmetro social = 1,9

C3 – parâmetro de congregação passiva seletiva = 1,9

Para todos os tipos de PSO avaliados os valores de velocidade e de posição da partícula serão limitados aos seguintes valores:

- Velocidade máxima e mínima (V_{max} e V_{min}): limitadas aos limites máximo e mínimo do espaço de busca de cada função; e

- Posição da partícula (X): limitada aos limites máximo e mínimo do espaço de busca de cada função, com a velocidade assumindo o valor nulo ($V=0$).

6.4.2– Parâmetros de Controle do Teste

A avaliação de desempenho do algoritmo utilizará os seguintes parâmetros de controle:

Número máximo de execuções = 30

Número máximo de gerações = 2000

Número máximo de partículas = 100

Dimensionalidade das funções de teste = 30

6.5 – FUNÇÕES DE TESTE

A avaliação do desempenho do algoritmo será realizada em seis funções de teste conhecidas (“benchmark”). A função F1 é uma função unimodal e as funções F2 a F6 são funções multimodais. Segue-se uma apresentação das funções de teste.

F1 – Função Sphere

$$f(x) = \sum_{i=1}^N x_i^2 \quad (11)$$

F2 – Função Rosenbrock

$$f(x) = \sum_{i=1}^{N-1} (100 * (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (12)$$

F3 – Função Griewank

$$f(x) = \frac{\sum_{i=1}^N x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (13)$$

F4 – Função Rastrigin

$$f(x) = \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (14)$$

F5 – Função Ackley

$$f(x) = \sum_{i=1}^{N-1} (20 + e - 20e^{-0.2\sqrt{0.5(x_{i+1}^2 + x_i^2)}} - e^{0.5(\cos(2\pi x_{i+1}) + \cos(2\pi x_i))}) \quad (15)$$

F6 – Função Schwefel

$$f(x) = 418.9829N - \sum_{i=1}^N x_i \sin(\sqrt{|x_i|}) \quad (16)$$

A função Sphere é uma função unimodal simples, a função Rosenbrock é uma função multimodal simples com um mínimo local e um ótimo global. As funções Griewank e Ackley são consideradas funções multimodais simples e as funções Rastrigin e Schwefel são consideradas funções multimodais complexas, todas apresentando vários mínimos locais. A tabela 2 apresenta as características básicas das funções de teste utilizadas.

Função	Dimensão	Espaço de Busca	Fmin	Erro Admissível
F1	30	$[-100; 100]^n$	0	0,01
F2	30	$[-30; 30]^n$	0	100
F3	30	$[-600; 600]^n$	0	0,05
F4	30	$[-5,12; 5,12]^n$	0	100,0
F5	30	$[-32,768; 32,768]^n$	0	0,01
F6	30	$[-500; 500]^n$	0	100,0

Tabela 2 – Características básicas das funções de teste

O Apêndice A apresenta os gráficos 3D das funções de teste [ABS DA CRUZ, 2007].

6.6 – PROGRAMA DO EXPERIMENTO

O algoritmo PSO-CPS proposto foi elaborado na linguagem C++ e executado em ambiente operacional Windows XP - Pentium (R) 4 CPU 3,2 GHz em uma plataforma Microsoft Visual Studio 2005, sob um IDE Microsoft Visual Studio 2005.

6.7– CALIBRAÇÃO DO PARÂMETRO C3 DA PSO-CPS

			Valor Médio da Função					
C1	C2	C3	SPHERE	ROSENBROCK	GRIEWANK	RASTRIGIN	ACKLEY	SCHWEFEL
0,5	0,5	0,5	2,13e-12	31,52	7,14e-3	64,45	0,85	3731
1,0	1,0	1,0	6,35e-39	20,89	5,41e-3	46,2	2,91e-14	4189
1,5	1,5	1,5	1,01e-62	11,81	6,66e-3	30,48	2,41e-14	3451
1,6	1,6	1,6	4,26e-70	7,89	4,623-3	25,01	2,0e-14	3289
1,7	1,7	1,7	6,36e-80	8,41	4,35e-3	25,5	1,99e-3	3450
1,8	1,8	1,8	2,1e-90	5,76	5,09e-3	26,07	1,69e-14	3465
1,9	1,9	1,9	2,63e-92	5,1	6,16e-3	25,5	1,74e-14	3415
2,0	2,0	2,0	6,14e-81	5,67	6,24e-3	23,75	1,53e-14	3279
2,1	2,1	2,1	2,93e-68	8,13	5,41e-3	24,71	1,48e-14	3329
2,2	2,2	2,2	1,02e-55	11,48	6,16e-3	21,72	1,24e-14	3337
2,3	2,3	2,3	1,22e-44	14,95	6,40e-3	21,79	1,08e-14	3147

Tabela 3 – Valores da calibração de C3 da PSO-CPS

Os resultados encontrados para calibração idêntica de C1, C2 e C3 em funções como Sphere e Rosenbrock foram para o valor de 1,9. A função Griewank apresentou seu melhor valor para 1,5. A função Rastrigin atingiu seu melhor valor com 2,2 e as funções Ackley e Schwefel para o valor de 2,3. Os resultados foram bastante dispersos, entre as funções de teste, entretanto pode-se observar que entre os valores de melhor desempenho para as funções se concentraram no intervalo de 1,7 a 2,3, nesse intervalo as diferenças entre os valores são praticamente desprezíveis. O valor $C1=C2=C3= 1,9$ será padronizado como parâmetro da PSO-CPS para avaliação de desempenho do algoritmo com outros tipos de PSO.

6.8 – RESULTADOS EXPERIMENTAIS

	Média da Função e (Desvio-Padrão)					
ALGORITMO	SPHERE	ROSENBROCK	GRIEWANK	RASTRIGIN	ACKLEY	SCHWEFEL
PSO-P	2,74e-85	11,13	20,04e-3	83,35	0,67	3925
	(1,12e-84)	(13,77)	(21,22e-3)	(26,26)	(0,68)	(309,8)
PSO-C	9,89e-73	1,2	19,06e-3	140,2	4,42	5000
	(5,04e-72)	(1,87)	(27,15e-3)	(30,9)	(2,96)	(468,9)
PSO-CP	1,98e-21	23,32	9,76e-3	85,7	1,23	4682
	(9,94e-21)	(3,03)	(11,8e-3)	(24,78)	(0,70)	(426,8)
PSO-CPS	2,63e-92	5,1	6,16e-3	25,5	1,74e-14	3415
	(4,60e-92)	(3,35)	(6,84e-3)	(8,09)	(4,64e-15)	(547,9)

Tabela 4 – Resultados Experimentais para as Funções de Teste

	Taxa de Sucesso					
ALGORITMO	SPHERE	ROSENBROCK	GRIEWANK	RASTRIGIN	ACKLEY	SCHWEFEL
PSO-P	100 %	100 %	86,7 %	80,0 %	50,0 %	0 %
PSO-C	100 %	100 %	86,7 %	3,33 %	0 %	0 %
PSO-CP	100 %	100 %	100 %	83,3 %	13,3 %	0 %
PSO-CPS	100 %	100 %	100 %	100 %	100 %	0 %

Tabela 5 – Taxa de Sucesso para as Funções de Teste

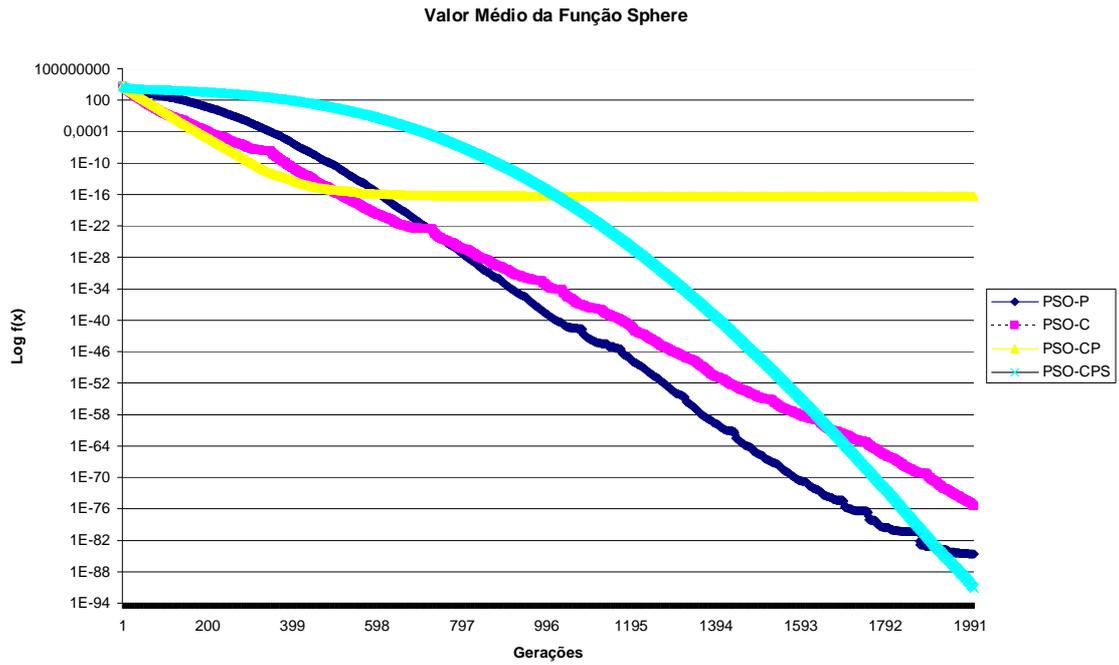


Figura 8 – Gráfico de valor médio da função Sphere

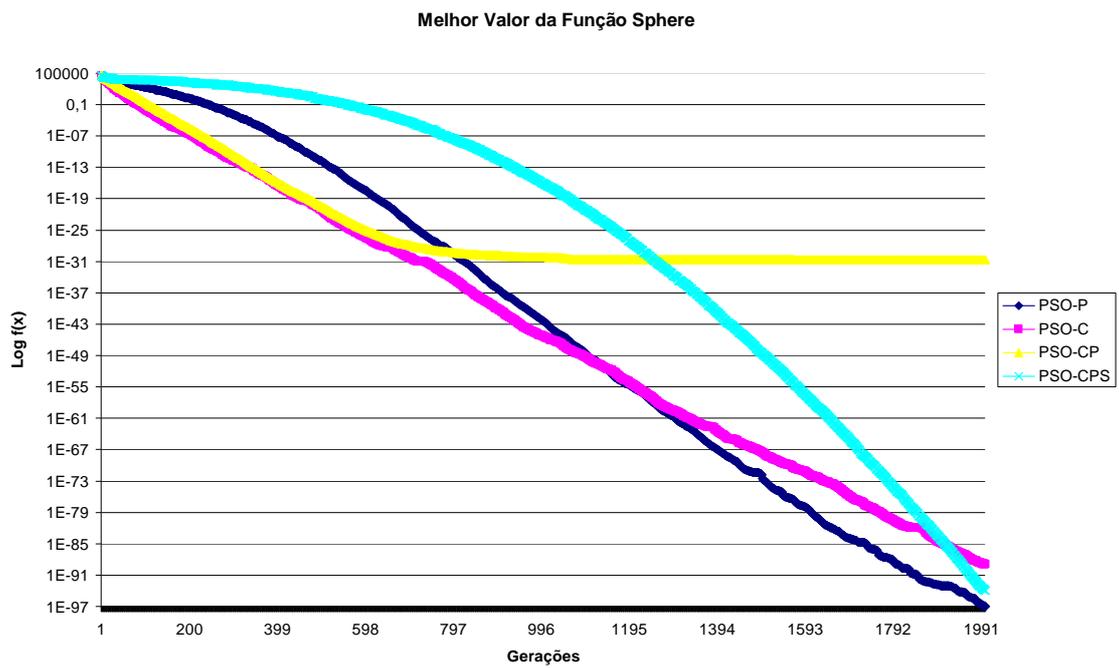


Figura 9 - Gráfico de melhor valor da função Sphere

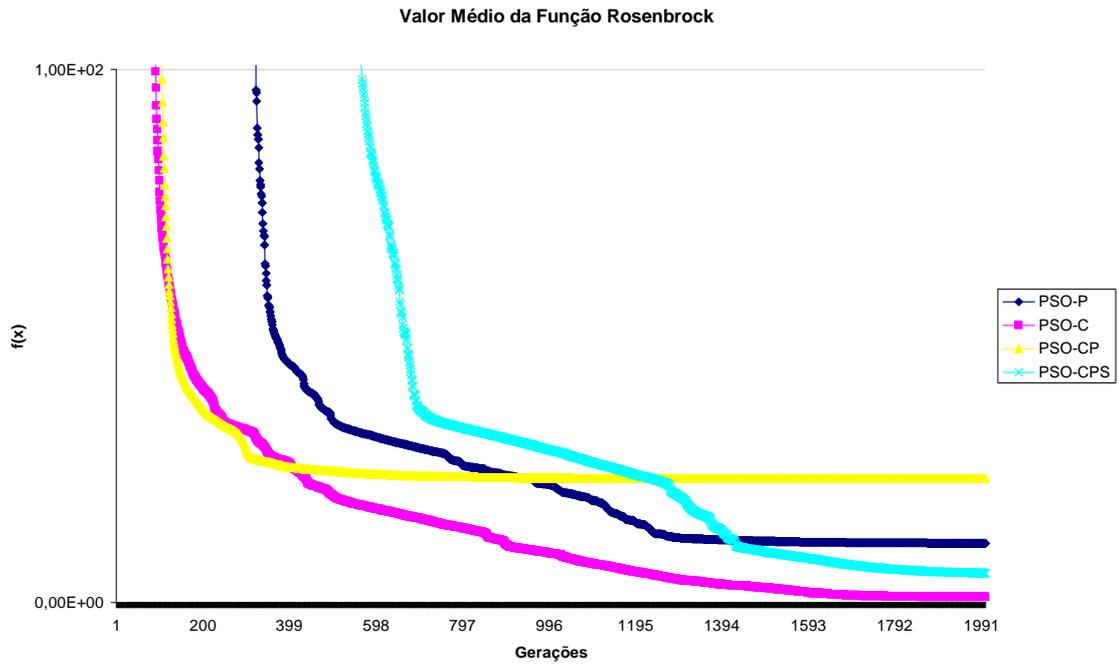


Figura 10 – Gráfico de valor médio da função Rosenbrock

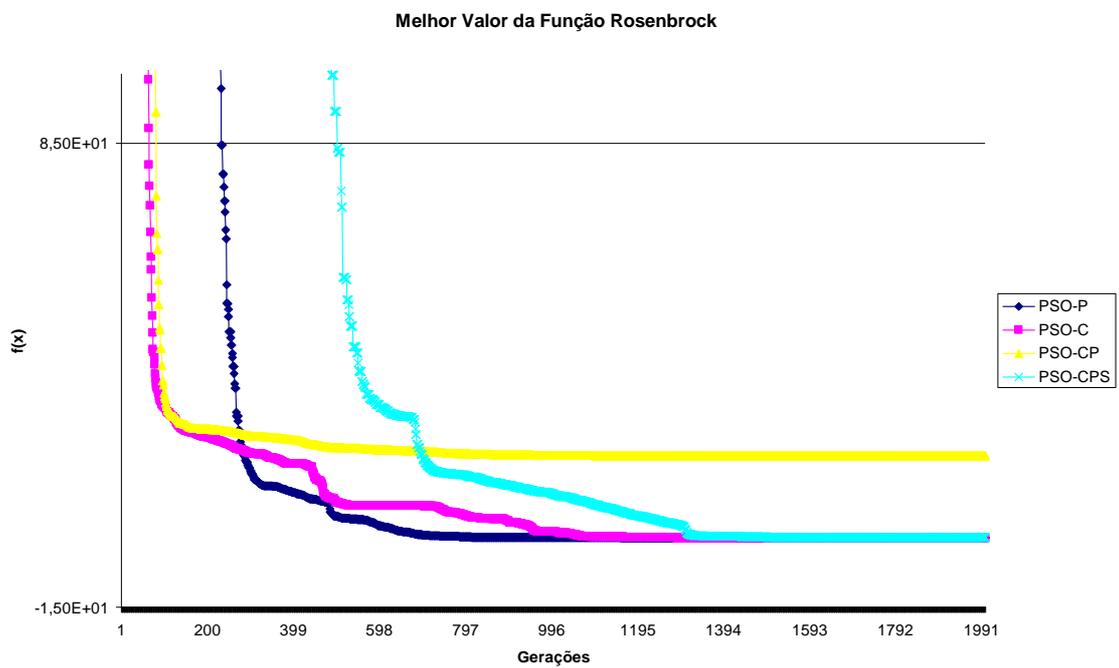


Figura 11 – Gráfico de melhor valor da função Rosenbrock

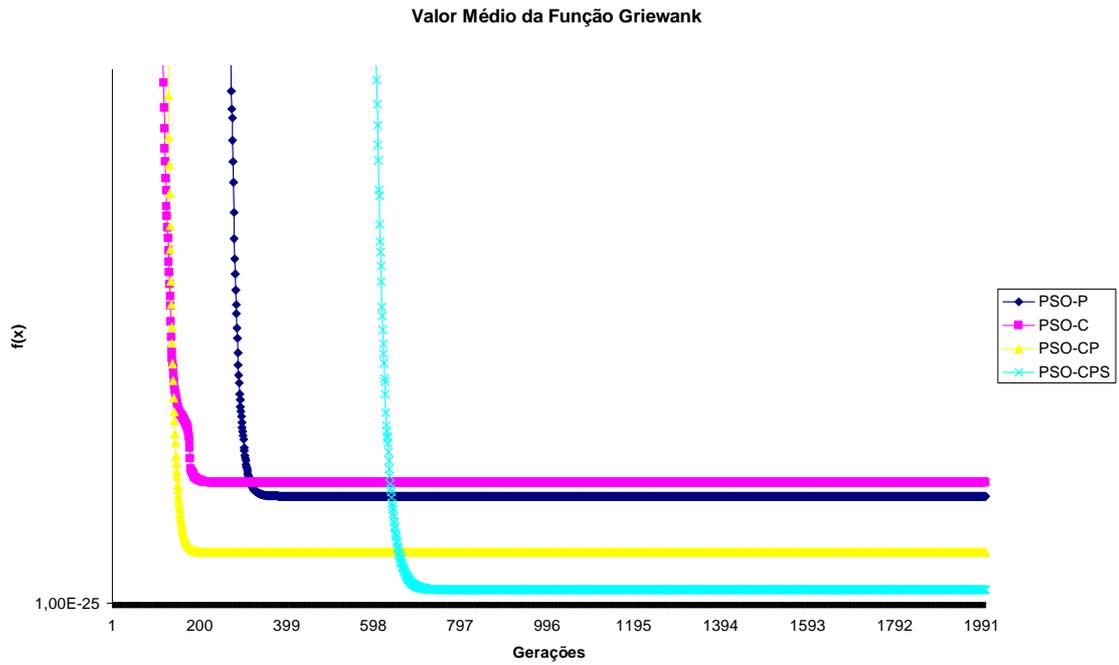


Figura 12 – Gráfico de valor médio da função Griewank

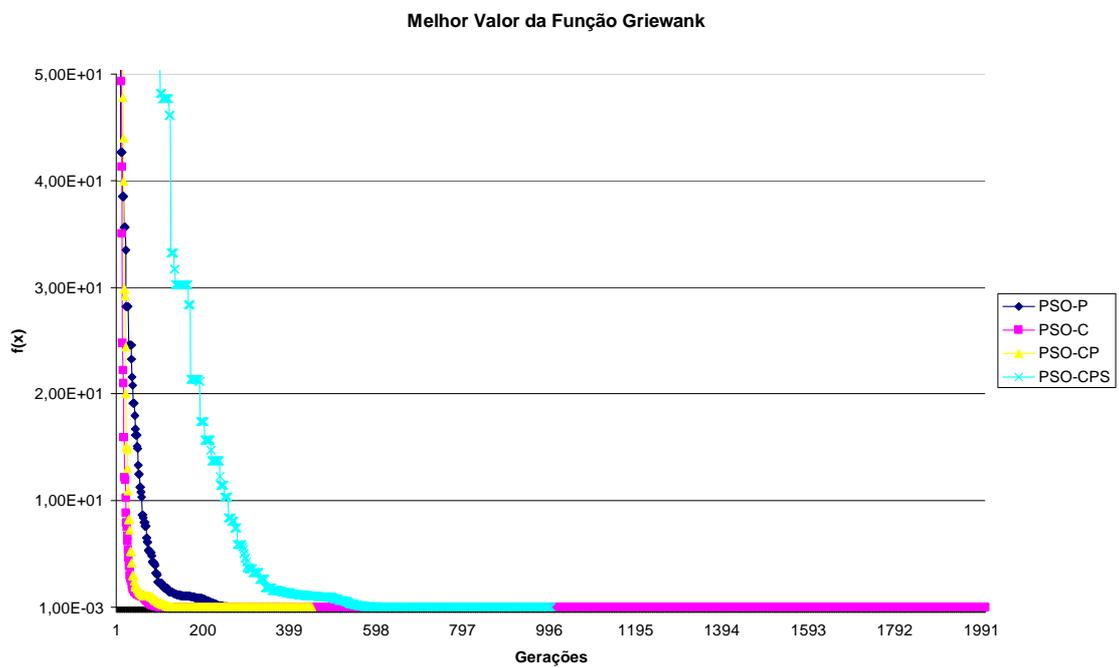


Figura 13 – Gráfico de melhor valor da função Griewank

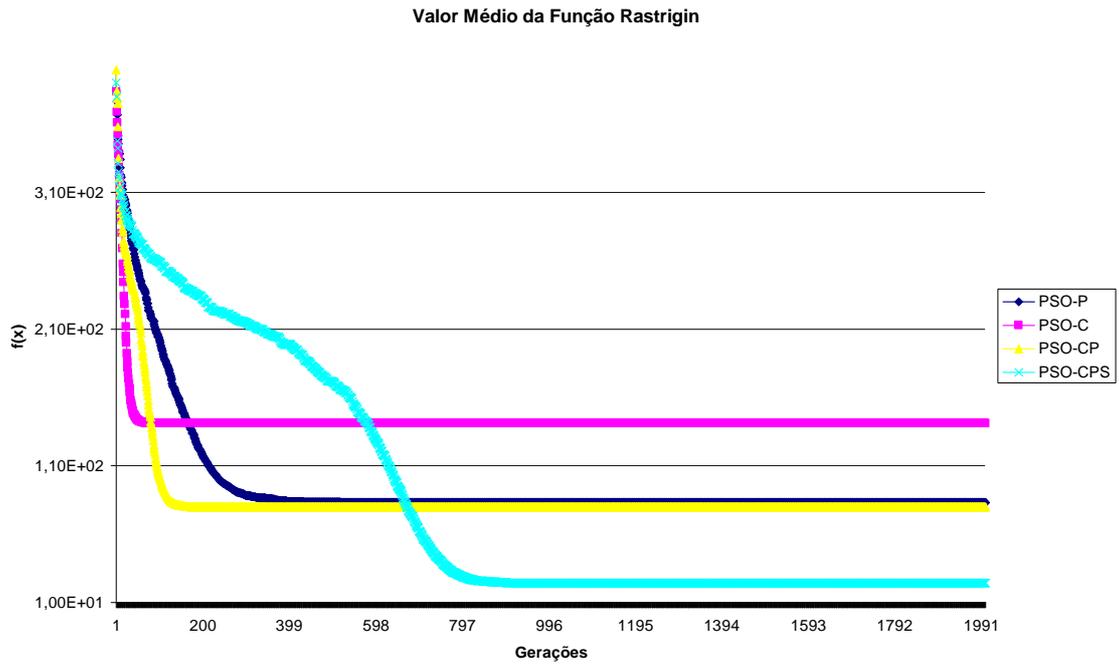


Figura 14– Gráfico de valor médio da função Rastrigin

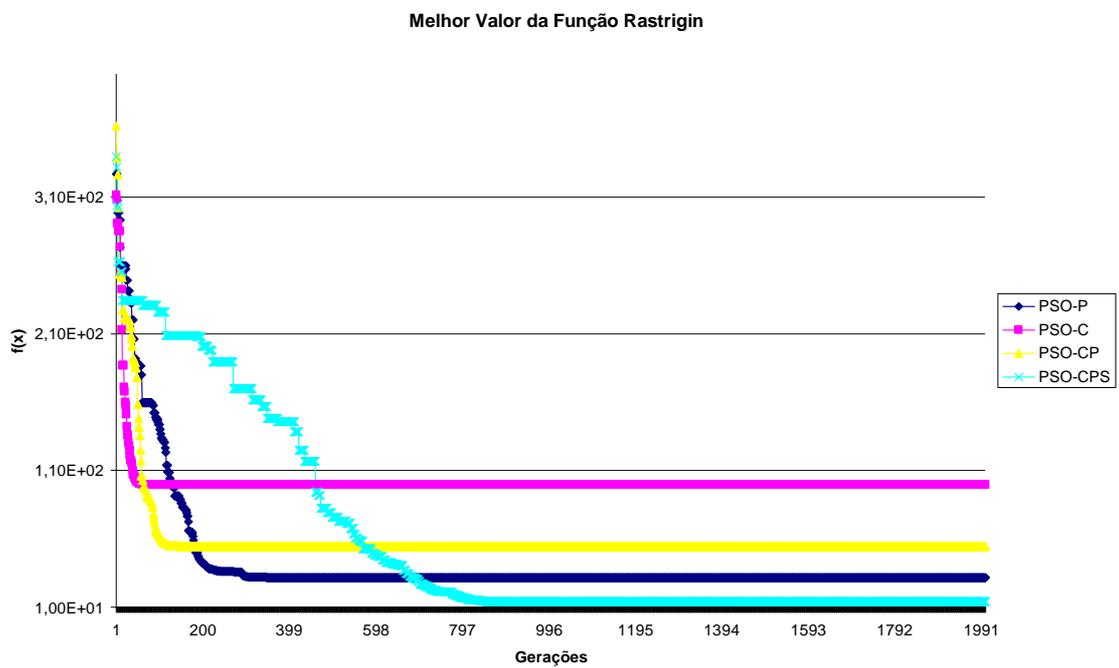


Figura 15 – Gráfico de melhor valor da função Rastrigin

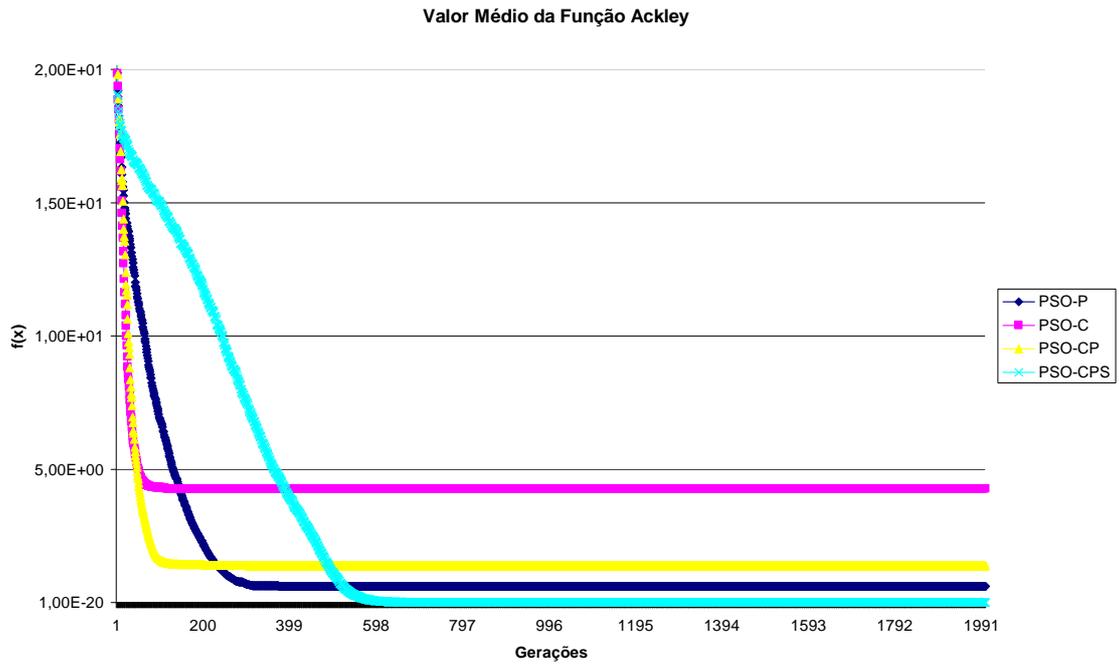


Figura 16 – Gráfico de valor médio da função Ackley

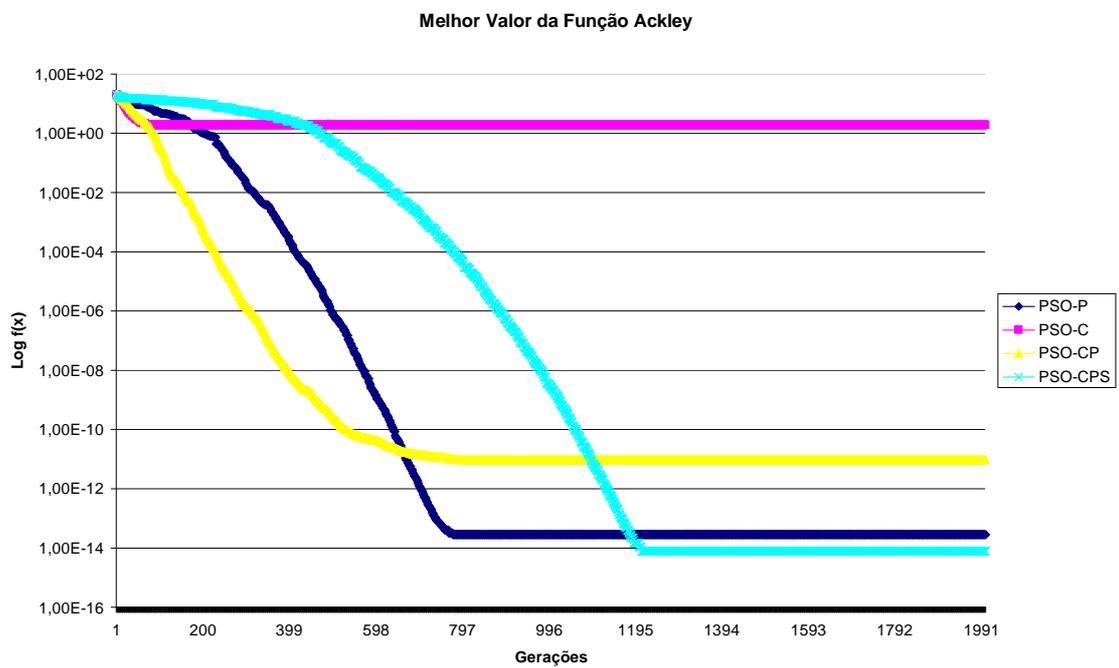


Figura 17 – Gráfico de melhor valor da função Ackley

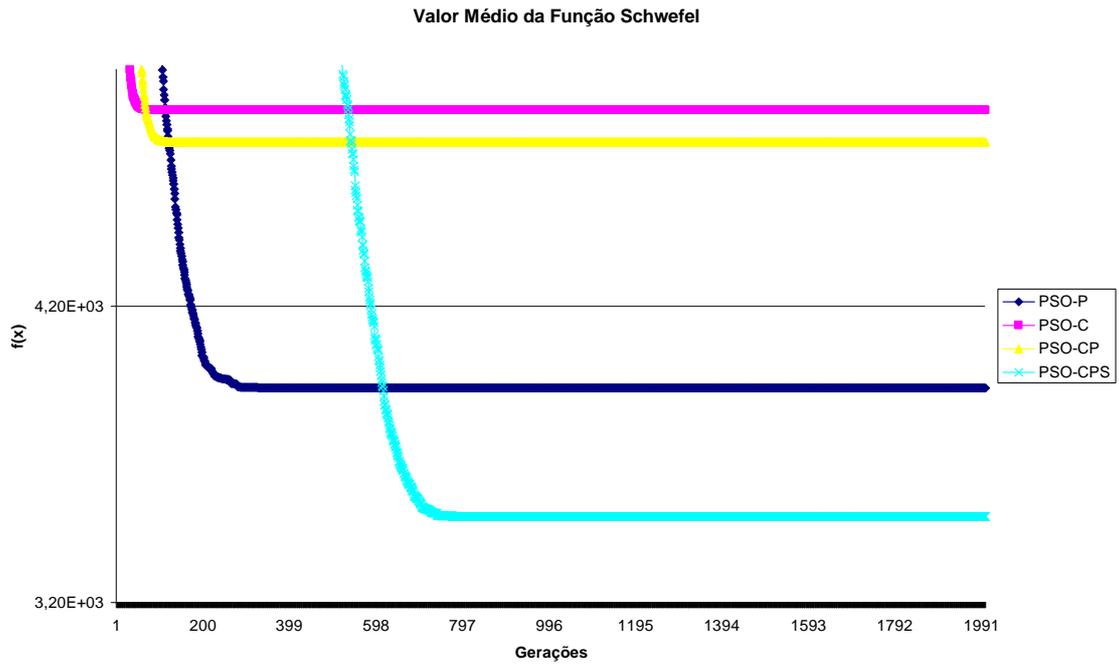


Figura 18 – Gráfico de valor médio da função Schwefel

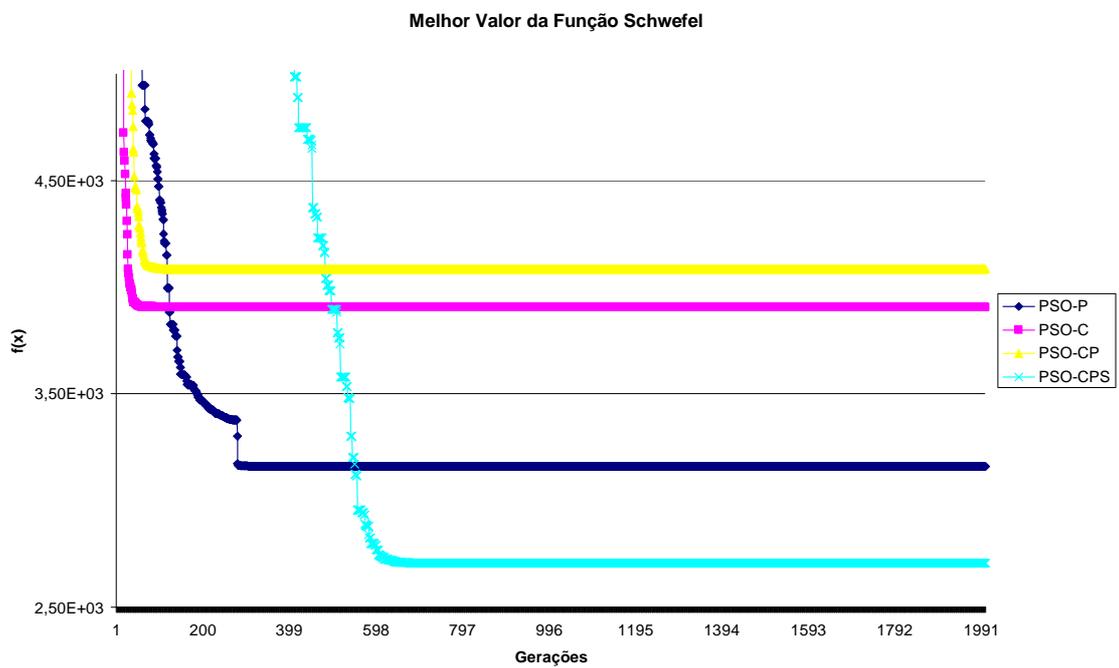


Figura 19 – Gráfico de melhor valor da função Schwefel

6.9 – ANÁLISE DE RESULTADOS

Os resultados experimentais, para avaliação de desempenho do algoritmo PSO-CPS que utiliza a congregação passiva seletiva em relação ao algoritmo da PSO-CP de congregação passiva aleatória, indicam a superioridade na resolução de problemas unimodais simples e problemas multimodais simples e complexos de todas as funções de teste utilizadas. Em comparação com os outros tipos de PSO (PSO-P Padrão, PSO-C Clerck e PSO-CP) o algoritmo PSO-CPS de congregação passiva seletiva foi superior nas funções de teste Griewank, Rastrigin, Ackley e Schwefel. Apenas na função Rosenbrock o desempenho foi inferior ao tipo de PSO-C (Clerc) cujo desempenho é particularmente adequado para essa função.

A tabela 5, de taxa de sucesso para as funções de teste, confirma a superioridade da PSO-CPS sobre as os demais tipos de PSO empregados. O resultado nulo para a função Shcwefel, para todos os tipos de PSO, deve-se à adoção de um erro máximo admissível igual a 100,0. Os melhores resultados médios encontrados para a função Schwefel foram: PSO-P = 3925, PSO-C = 5000, PSO-CP = 4682 e PSO-CPS = 3415. Apesar do erro máximo admissível não ter sido atingido, a PSO-CPS também apresentou o melhor resultado entre os tipos de PSO avaliados.

De acordo com os resultados da taxa de sucesso é possível categorizar os diversos tipos de PSO em função da dificuldade de otimização das funções de teste. As funções simples Sphere e Rosenbrock, unimodal e multimodal respectivamente foram facilmente otimizadas. Já os mínimos locais das funções multimodais foram altamente prejudiciais ao desempenho dos PSO-P (Padrão) nas funções Griewank (86,7%), Rastrigin (80%) e Ackley (50%), ao desempenho da PSO-C (Clerc) nas funções Griewank (86,7%), Rastrigin (3,3%) e Ackley (0%) e no desempenho da PSO-CP (Congregação Passiva) nas funções Rastrigin (83,3%) e Ackley (13,3%).

De acordo com os resultados dos testes experimentais para as funções Sphere, Rosenbrock, Griewank, Rastrinhin e Ackley, apenas o PSO-CPS não foi afetado pelos mínimos locais dessas funções, mostrando-se robusto no combate à convergência prematura.

Os resultados da função Schwefel indicam que todos os tipos de PSO estudados foram atraídos por seus mínimos locais. Esse fato também foi observado em outros estudos como o de Godoy e Von Zuben [GODOY & VON ZUBEN, 2009] onde o valor mínimo atingido foi de 4160.

No presente trabalho o desempenho dos 4 tipos de PSO testados pode ser resumido pela seguinte ordenação:

$$\text{PSO-CPS} > \text{PSO-CP} > \text{PSO-P} > \text{PSO-C}$$

6.9.1 – Complementação de Resultados

Como complementação aos resultados de desempenho encontrados para o algoritmo PSO-CPS, a tabela 6 apresenta uma comparação do algoritmo PSO-CPS com os resultados recentemente publicados no trabalho de Godoy e VonZuben [GODOY e VON ZUBEN, 2009] para o algoritmo CNPSO (alínea f do item 5.4.2). Para essa comparação, na execução do algoritmo PSO-CPS foram utilizados os mesmos parâmetros de teste do referido trabalho, ou seja:

- Número máximo de execuções = 30
- Número máximo de gerações = 8000
- Número máximo de partículas = 200
- Dimensionalidade das funções de teste = 30

ALGORITMO	Média da função (Desvio-Padrão)	
	CNPSO	PSO-CPS
SPHERE	5,31e-64	3,23e-86
	(1,41e-63)	(1,58e-85)
ROSENBROCK	19,3	1,40
	(22,3)	(1,93)
GRIEWANK	5,25e-3	7,06e-3
	(7,21e-3)	(7,93e-3)
RASTRIGIN	34,8	40,59
	(6,69)	(12,69)
ACKLEY	8,31e-1	3,15e-14
	(3,64)	(1,43e-14)
SCHWEFEL	4160	3497
	(471,0)	(658,7)

Tabela 6 – Comparação dos algoritmos CNPSO x PSO-CPS

De acordo com a tabela 6 o algoritmo PSO-CPS possui desempenho superior ao algoritmo CNPSO nas funções Sphere, Rosenbrock, Ackley e Schwefel e apresenta para as funções Griewank e Rastrigin valores inferiores, porém na mesma ordem de grandeza.

7 - CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS

O conceito de melhoria contínua do processo de movimentação das partículas e a proposta pela utilização de partículas vizinhas somente com desempenho superior ao da própria partícula mostraram-se corretos uma vez que os resultados da PSO-CPS foram superiores, em todas as funções de teste, aos valores calculados para a PSO-CP.

A convergência prematura observada nos demais tipos de PSO avaliados não esteve presente no desempenho do algoritmo PSO-CPS. Em todas as funções de teste, a fase de busca local do algoritmo proposto ocorreu sempre posteriormente aos outros algoritmos impedindo que o algoritmo PSO-CPS fosse prematuramente capturado pelos mínimos locais das funções multimodais estudadas.

Como sugestão para os próximos trabalhos e seguindo a linha de soluções simples e de baixo custo computacional acredita-se que os seguintes estudos possam aumentar o desempenho do algoritmo PSO-CPS:

- a) Estudo da calibração dos parâmetros cognitivo (C1), parâmetro social (C2) e parâmetro de congregação passiva seletiva (C3) para valores distintos.
- b) Verificação do desempenho do algoritmo de acordo com novos valores de parâmetros do teste, ou seja: tamanho da população, número máximo de gerações e dimensionalidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABS DA CRUZ, A. V., **Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica**, Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, 2007.
- ANGELINE, P. J., **Evolutionary optimization versus Particle Swarm Optimization: Philosophy and performance differences**. In: Porto VW, Saravanan N, Waagen D and Eiben AE (eds) *Evolutionary Programming VII*, pp. 601–610. Springer, 1998.
- BÄCK, T., FOGEL, D. B., & MICHALEWICZ Z., **Handbook of evolutionary computation**. Bristol, Philadelphia: Institute of Physics Publishing. New York, Oxford: Oxford University Press, 1997.
- BARABÁSI, A. L., ALBERT, R., **Emergence of Scaling in Random Networks**. *Science*, 286(5439): pp 509–512, 1999.
- BAZARAA, M. S.; SHETTY, C. M., **Nonlinear Programming - Theory and Algorithms**. John Wiley & Sons, New York, 1979.
- BERGH, F., ENGELBRECHT, A., **A Cooperative Approach to Particle Swarm Optimization**, *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, 2004.
- BONABEAU E.; DORIGO, M.; THERAULAZ, T., **Swarm Intelligence: From Natural to Artificial Systems**, New York: Oxford University Press. 1999.
- BULLNHEIMER, B., HART, R. F., STRAUS, C., **Applying the Ant System to the vehicle routing problem, in: Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization**, eds. S. Voss, S. Matello, L. H. Osman and C. Roucariol, Kluwer, Boston, 1998.
- CARLISLE, A., DOZIER, G., **An Off-The-Shelf PSO**. *Proceedings of the Particle Swarm, Optimization Workshop*, pp. 1–6, 2001.

CHATTERJEE, A., SIARRY, P. **Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization**, Computers & Operational Research, Volume 33, Issue 3, pp 859-871, 2006.

CLERC, M., **When Nearer is Better**. Draft 2007-05-18, 2007, disponível em:
<http://hal.inria.fr/docs/00/15/20/53/PDF/NisBetter2.pdf>

CLERC, M., **Particle Swarm Optimization**. International Scientific and Technical Encyclopedia - ISTE, 2006.

CLERC, M., **The swarm and the queen: towards a deterministic and adaptive particle swarm optimization**. Proc. 1999 ICEC, Washington, DC, pp 1951-1957, 1999.

COELHO, L. S. & COELHO, A. A. R., **Algoritmos evolutivos em identificação e controle de processos: uma visão integrada e perspectivas**. Revista SBA Controle & Automação, 10(1), pp 13-30, 1999.

COELHO, L. S., **Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos** - São Carlos, SP : SBMAC XII, 103 p. - (Notas em Matemática Aplicada; 2), 2003

CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., **Algoritmos: Teoria e Prática**. Tradução da 2ª edição Americana por Vanderberd D. de Souza – Rio de Janeiro: Campus, 2002, pp 763-807, 1999.

DARWIN, C., **Origin of species by means of natural selection, or the preservation of favored races in the struggle for life**. 6th Edition, v. I and II. John Murray : London, Albemarle Street, 1859 (1a. ed), disponível em:
<http://honors.ccsu.ctstateu.edu/Honors/EText/Darwin/DarwinOriginContents.html>

DEL VALLE, YAMILLE., VENAYAGAMOORTHY. G. K., MOHAGHEGHI, S., HERNANDEZ, J. C., HARLEY R. G., **Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems**, IEEE Transactions

on Evolutionary Computation, 12 (2) pp 171-195, 2008.

DENEUBOURG, J. L., ARON, S., GOSS, S., PASTEELS, J. M., **Self-organized shortcuts in the Argentine ant.** Naturwissenschaften, 76: pp 579-581, 1989.

DORIGO M., Di Caro G., Gambardella L.M., **Ant Algorithms for Discrete Optimization**, Artificial Life, Vol. 5, N. 2, 1999.

DORIGO, M., GAMBARDELLA, L. M., **Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem.** IEEE Transactions on Evolutionary Computation, 1 (1) pp 53-66, 1997.

DORIGO, M., MANIEZO, V., COLORNI, A., **The Ant System: Optimization by a Colony of Cooperating Ants.** IEEE Transactions on Systems, Man, and Cybernetics, Part B, 26, pp 1-13. 1996.

DORIGO, M. MANIEZO, V., COLORNI, A., **Ant System: An Autocatalytic Optimizing Process, Technical Report**, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, 1991.

DRÉO, J., SIARRY, P., **Swarm Intelligence, Focus on Ant and Particle Swarm Optimization**, Itech Education and Publishing, Viena, pp. 532, 2007

EBRHART, R. C., SHI, Y., **Parameter Selection in Particle Swarm Optimization.** In Evolutionary Programming VII, pp 591- 616. Lecture Notes in Computer Science 1447, 1998.

EBERHART, R. C., SIMPSON, P., DOBBINS, R., **Computational Intelligence PC Tools**, Academic Press, 1996.

EBERHART, R. C., KENNEDY, J., **A New Optimizer Using Particle Swarm Theory.** in “Proceedings of the Sixth International Symposium on Micro Machine and Human Science”, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp.39-43, 1995.

- EBERHART, R. C., SHI, Y., **Particle Swarm Optimization: Developments, Applications and Resources**, in Proc. IEEE Congr. Evol. Comput., vol. 1, pp. 81–86, 2001.
- EL-DIB, A., YOUSSEF, H., EL-METWALLY, M., OSMAN, Z., **Load Flow Solution Using Hybrid Particle Swarm Optimization**, in Proc. Int. Conf. Elect., Electron., Comput. Eng., pp. 742–746, 2004.
- FAYYAD U., PIATESTKU-SHAPIO G., SMYTH P., UTHURUSAMY R., **Advances in Knowledge Discovery and Data Mining**, AAAI/MIT Press, 1996.
- FEO, T., and RESENDE, M. G., **Greedy Randomized Adaptive Search Procedures**. Journal of Global Optimizations 6: 109-13. 1995.
- FOGEL, D. B., **An Introduction to Simulated Evolutionary Optimization**. IEEE Transactions on Neural Networks, 5(1), 3-14, 1994.
- GAMBARDELLA, L. M., DORIGO, M. , **An Hibrid Ant System for the Sequential Ordering Problem**. Tchenical report IDSIA-11-97, IDISIA, Lugano, Switzerland, 1997.
- GAMBARDELLA, L. M., TAILLARD, E., DORIGO, M., **Ant Colonies for the QAP**. Technical Report IDISIA-4-97, IDISIA, Lugano, Switzerland, 1997.
- GAMBARDELLA, L. M., TAILLARD, E., AGAZZI, G., **Ant Colonies for Vehicle Routing Problems**. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas for Optimization*, McGraw-Hill, 1999.
- GODOY, A., Godoy VON ZUBEN, F. J., **A Complex Neighborhood Based Particle Swarm Optimization**. In Congress on Evolutionary Computation (CEC 2009), pp 720-727, 2009.
- GROSAN, C., ABRAHAM, A., CHIS, M., **Swarm Intelligence in Data Mining**, Studies in Computational Inteligence (SCI), Volume 3, pp 1-16, 2006.

- GLOVER, F., **Tabu Search – Part 1**. ORSA Journal of computing. 1, 190-206, 1989.
- GLOVER, F., **Tabu Search – Part 2**. ORSA Journal of computing. 2, 4-32, 1990.
- GOLDBERG, D. E., **Genetic algorithms in search, optimization, and machine learning**. Addison Wesley: Reading, USA, 1989.
- HAFTKA, R. T., KAMAT, M. P., **Elements of Structural Optimization**. Martinus Nijhoff Publishers, Dordrecht, 1985.
- HE, S., WEN, J., PREPAIN, E., WU, Q., FITCH, J., MANN S., **An Improved Particle Swam Optimization for Optimal Power Flow, in Proc. Int. Conf. Power Syst. Technol., pp. 1633–1637, 2004.**
- HOLLAND, J., **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.
- HU, X., **Particle Swarm Optimization**, in Tutorial of the IEEE Swarm Intell. Symp., 2006.
- HU, X. EBERHART, R., **Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems**, in Proc. IEEE Congr. Evol. Comput., vol. 2, pp. 1666–1670, 2002.
- JANSON, S., MIDDENDORF, M., **A Hierarchical Particle Swarm Optimizer and its Adaptive Variant**. IEEE Transactions on System Man and Cybernetics B, 35(6), pp: 1272–1282, 2005.
- KANNAN, S., SLOCHANAL, S., PADHY, N., **Application of Particle Swarm Optimization Technique and its Variants to Generation Expansion Problem**, ELSERVIER Electric Power Syst. Res., vol. 70, no. 3, pp 203–210, 2004.
- KENNEDY, J., MENDES, R., **Population Structure and Particle Swarm Performance**. In Proceedings of the IEEE Congress on Evolutionary

Computation (CEC), 1671-1676, 2002.

KENNEDY, J., EBERHART, R. C., **Swarm Intelligence**, Morgan Kaufmann Publishers, ISBN: 1-55860-595-9, 2001.

KENNEDY, J., **Stereotyping: Improving Particle Swarm Performance with Cluster Analysis**, Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, CA, 2000, pp. 1507–1512, San Diego, CA, 2000.

KENNEDY, J., **Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance**. Proceedings of the Congress of Evolutionary Computation, pp. 1931–1938, Washington D.C., 1999.

KENNEDY, J. and EBERHART, R., **Particle swarm optimization**.
In Proceedings of ICNN'95 - International Conference on Neural Networks, volume 4, pp 1942-1948, 1995.

KIRKPATRICK, S., GELATT Jr, C. D., VECCHI, M. P., **Optimization by Simulated Annealing**. Science, 220 (4598), 1983.

KIRSCH, U., **Optimum Structural Design**. McGraw-Hill, 1981.

LIANG, J., QIN, A., SUGANTHAN P., BASKAR S., **Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions**,”IEEE Trans. Evol. Comput., vol. 10, no. 3, pp.281–295, 2006.

LUENBERGER, D. G., **Introduction to Nonlinear Programming**. Addison-Wesley Publishing Company, London, 1984.

MANIEZZO, V., COLORNI, A., **The ant system applied to the quadratic assignment problem**, IEEE Transactions on Knowledge and Data Engineering, 11(5), 769-778, 1999.

MENDES, R, **Population Topologies and Their Influence in Particle Swarm**

- Performance.** Tese de Doutorado, Departamento de Informática - Escola de Engenharia – Universidade do Minho, 1994.
- NAKA, S., GENJI, T., YURA, T., FUKUYAMA, Y., **A Hybrid Particle Swarm Optimization for Distribution State Estimation**, IEEE Trans. Power Syst., pp. 60–68, 2003.
- NEVES, F. A., **Programação com Multi-Objetivos Aplicada à Otimização do Projeto de Pontes Estaiadas.** Tese de Doutorado na COPPE / UFRJ, Rio de Janeiro, 1997.
- PARRISH, J. K., HAMNER, W. M., **Animal Groups in Three Dimensions**, Cambridge University Press, Cambridge, UK, 1997.
- PARSOPOULOS, K., and VRAHATIS, M., **Recent Approaches to Global Optimization Problems Through Particle Swarm**, In Natural Computing1: 232-306. Kluwer Academic Publishers, 2002.
- PARSOPOULOS, K., VRAHATIS, M., **Particle Swarm Optimization Method in Multiobjective Problems**, in Proc. ACM Symp. Appl. Comput., pp. 603–607, 2002.
- PERAM, T., VEERAMACHANENI, K., MOHAN, C., **Fitness-Distance Ratio based Particle Swarm Optimization.** In Proceedings of the Swarm Intelligence Symposium (SIS) – IEEE2003, pp. 174–181, Indianapolis, Piscataway, 2003.
- POLI, R., KENNEDY, J., BLACKWELL, T., **Particle Swarm Optimization – An overview**, Swarm Intelligence Vol. 1 N 1, pp 33-57, 2007.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. and FLANNERY, B. P., **Numerical Recipes in Fortran.** University of Cambridge, 1992.
- RECHENBERG, I. **Evolution strategy.** Computational Intelligence – Imitating Life, 147-159, IEEE Press, 1994.

- REYNOLDS, C. W., **Flocks, Herds and Schools: A Distributed Behavioral Model**.
Computer Graphics, 21(4):25-34, 1987
- RICHARDS, M., VENTURA, D., **Choosing a Starting Configuration for Particle Swarm Optimization**, In Proceedings IEEE International Joint Conference on Neural Networks, Volume: 3: pp 2309-2312 – 2004.
- RICHARDS, M., VENTURA, D., **Dynamic Sociometry in Particle Swarm Optimization**, Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing, pp. 1557–1560, North Carolina – 2003.
- SCHACHTER, R, D, **Optimization Techniques with Knowledge Based Control in Ship Concept Design**, PHD Thesis, Department of Mechanical Engineering, Brunel University, UK, 1990.
- SECREST, B., LAMONT, G., **Visualizing Particle Swarm Optimization - Gaussian Particle Swarm Optimization**, in Proc. IEEE Swarm Intell. Symp., pp. 198–204, 2003.
- SHI, Y., Feature Article on Particle Swarm Optimization, IEEE Neural Network Society, Feature Article, pp. 8–13, 2004.
- SHI, Y., EBERHART, R. C., **Fuzzy Adaptive Particle Swarm Optimization**. In Proceedings of the IEEE Congress on Evolutionary Computation CEC2001, pp 101-106, 2001.
- SUGANTHAN, P. N., **Particle swarm optimiser with neighbourhood operator**. In Proceedings of the IEEE Congress on Evolutionary Computation CEC1999, Piscataway, pp. 1958–1962, 1999.
- VLACHOGIANNIS, J. G., KWANG Y. L., **A Comparative Study on Particle Swarm Optimization for Optimal Steady-State Performance of Power Systems**, IEEE Transactions on Power Systems, 21 (4), 2006.

WATTS, D. J., **Small Worlds: The Dynamics of Networks Between Order and Randomness**, Princeton University Press 1999.

WATTS, D. J., STROGATZ S. H., Strogatz, **Collective dynamics of ‘small-world’ networks**. Nature, 393, pp 440-442, 1998.

XIE, X. ZHANG, W., YANG Z., **A Dissipative Particle Swarm Optimization**, in Proc. IEEE Congr. Evol. Comput., vol. 2, pp. 1456–1461, 2002.

ZHANG, W., XIE, X., **DEPSO: Hybrid Particle Swarm with Differential Evolution Operator**, in Proc. IEEE Int. Conf. Syst., Man, Cybern., vol. 4, pp. 3816–3821, 2003.

ZHANG, W. LIU, Y., CLERC, M., **An Adaptive PSO Algorithm for Reactive Power Optimization**, in Proc. 6th Int. Conf. Advances in Power System Control, Operation and Management, pp. 302–307, 2003.

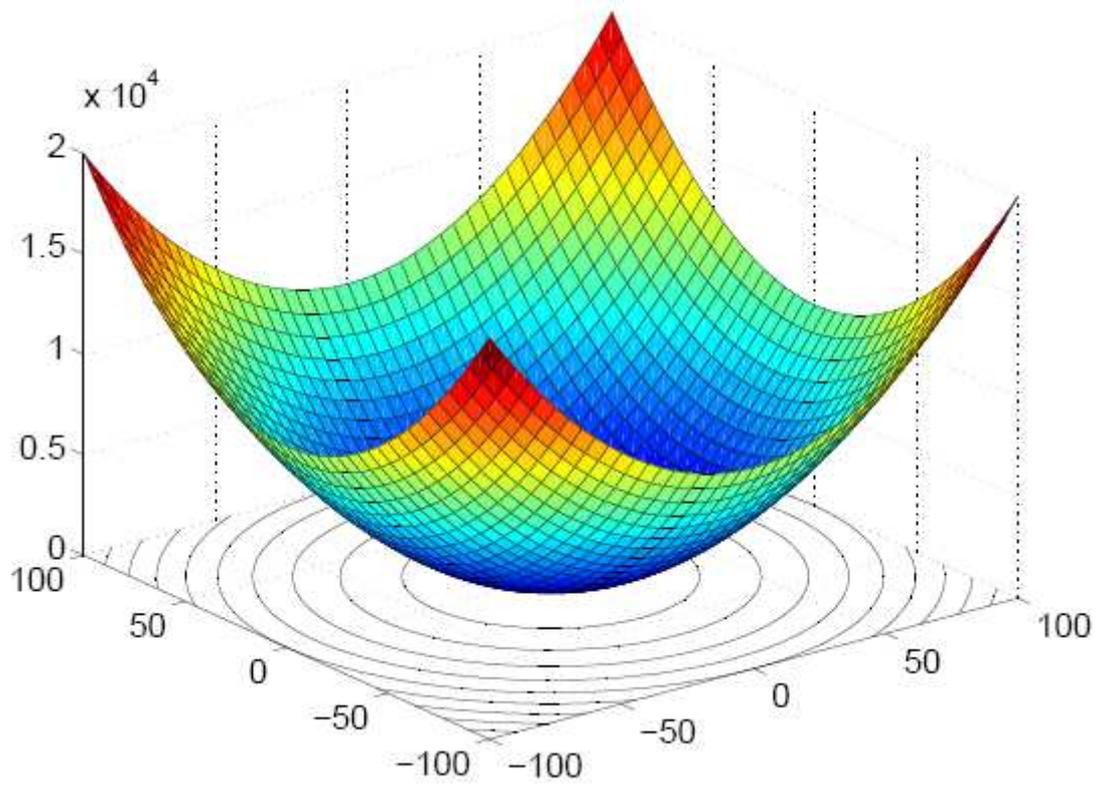
ZHIHUA, C., YONGFANG, C., XINGJUAN, C., **Nearest Neighbor Interaction PSO based on Small-World Model**, Lecture Notes in Computer Science, vol 5788, pp. 633-640, 2009.

APÊNDICE A

GRÁFICOS 3D DAS FUNÇÕES DE TESTE

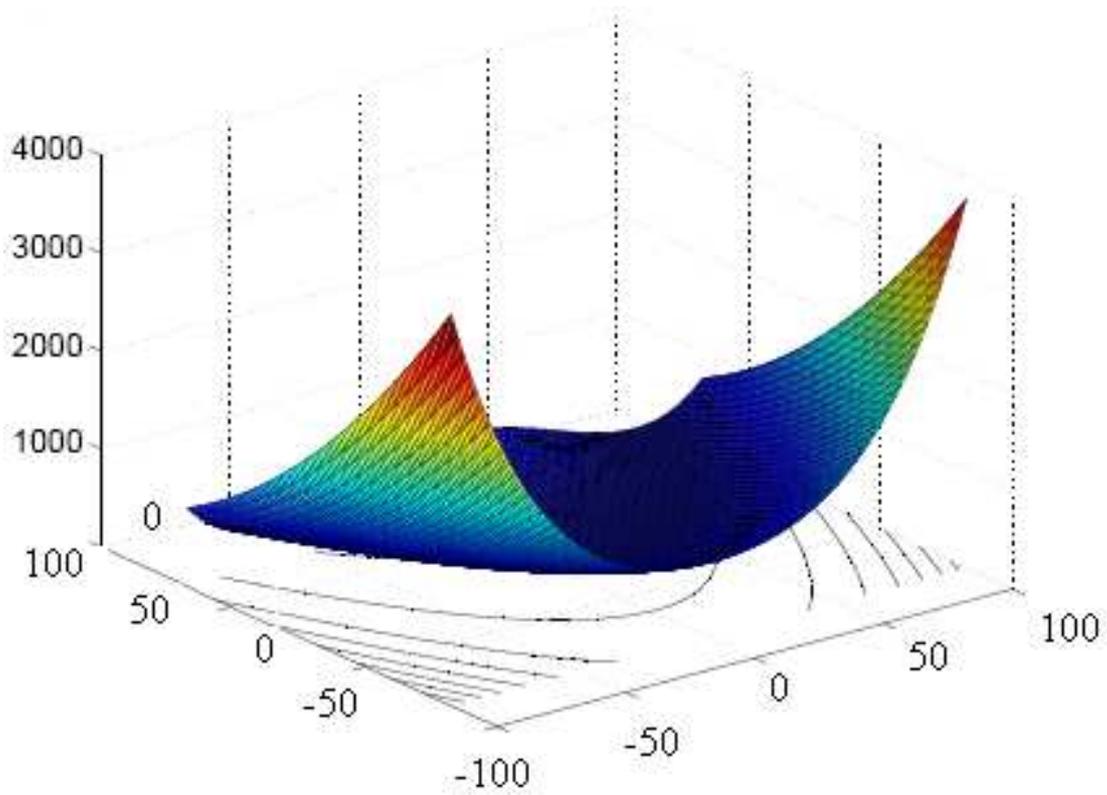
F1 – FUNÇÃO SPHERE

$$f_{Sphere}(\mathbf{x}) = \sum_{i=1}^D x_i^2$$



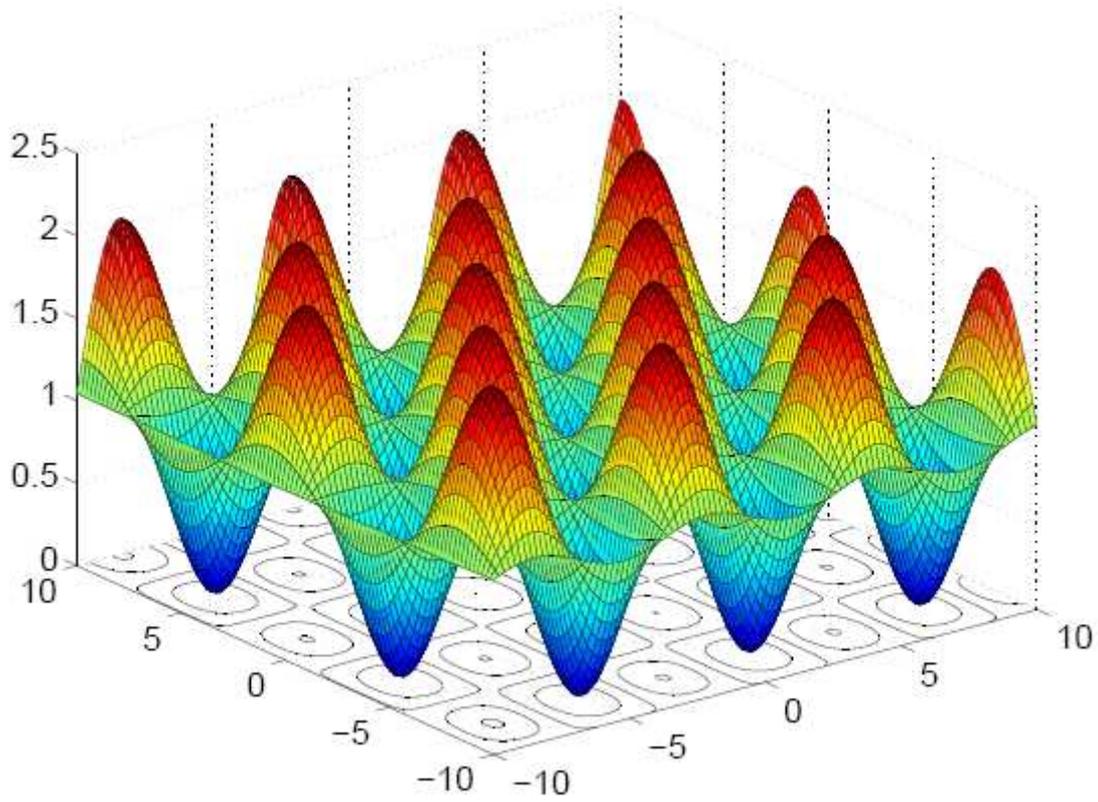
F2 – FUNÇÃO ROSENBROCK

$$f_{Rosenbrock}(\mathbf{x}) = \sum_{i=1}^{D-1} \left[100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$$



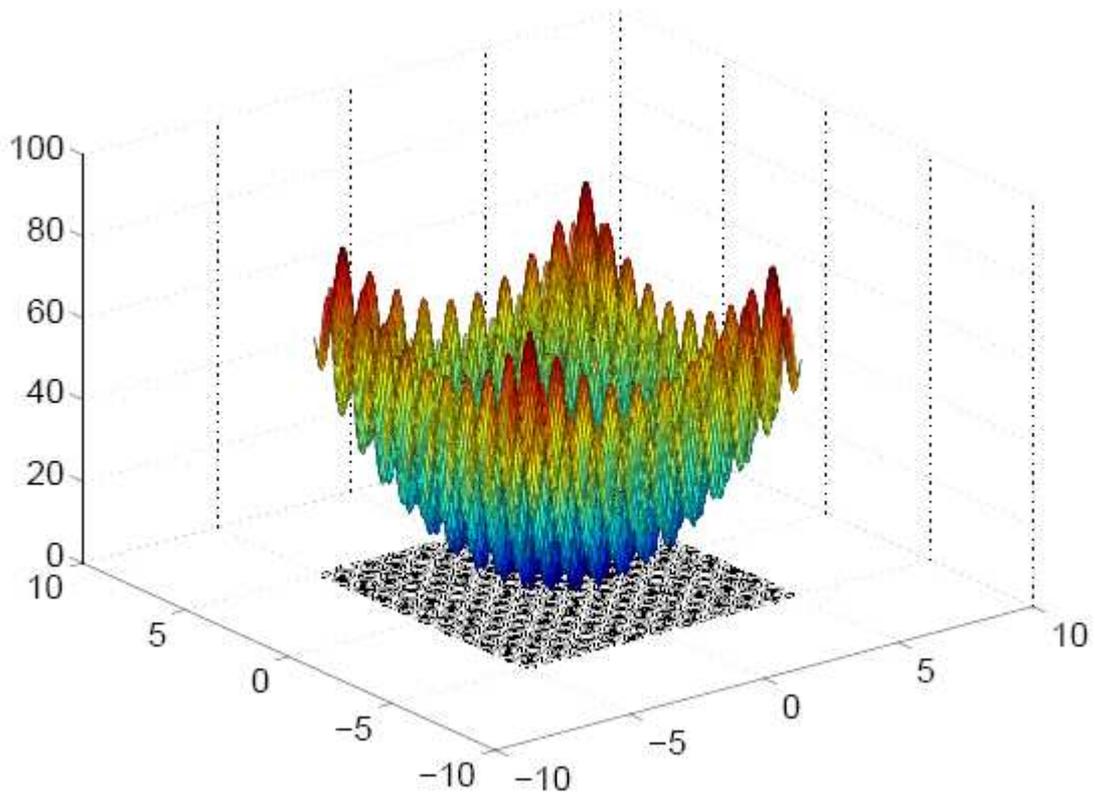
F3 – FUNÇÃO GRIEWANK

$$f_{Griewank}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$



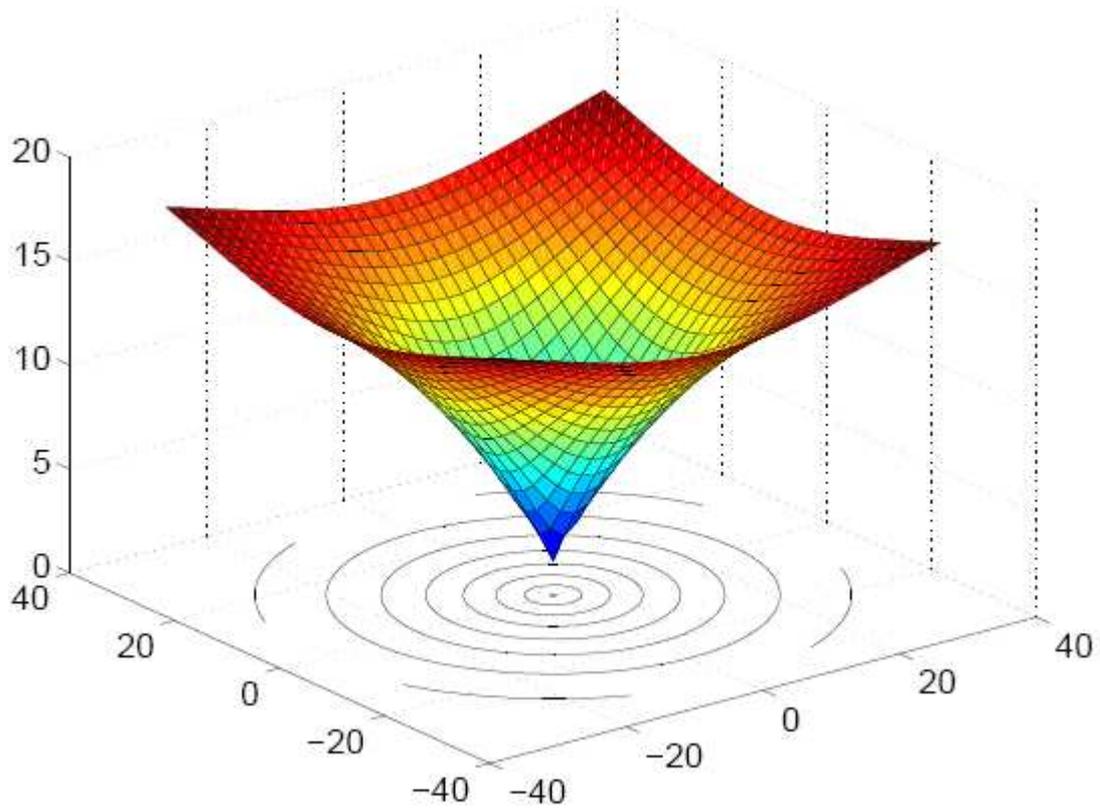
F4 – FUNÇÃO RASTRIGIN

$$f_{\text{Rastrigin}}(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$$



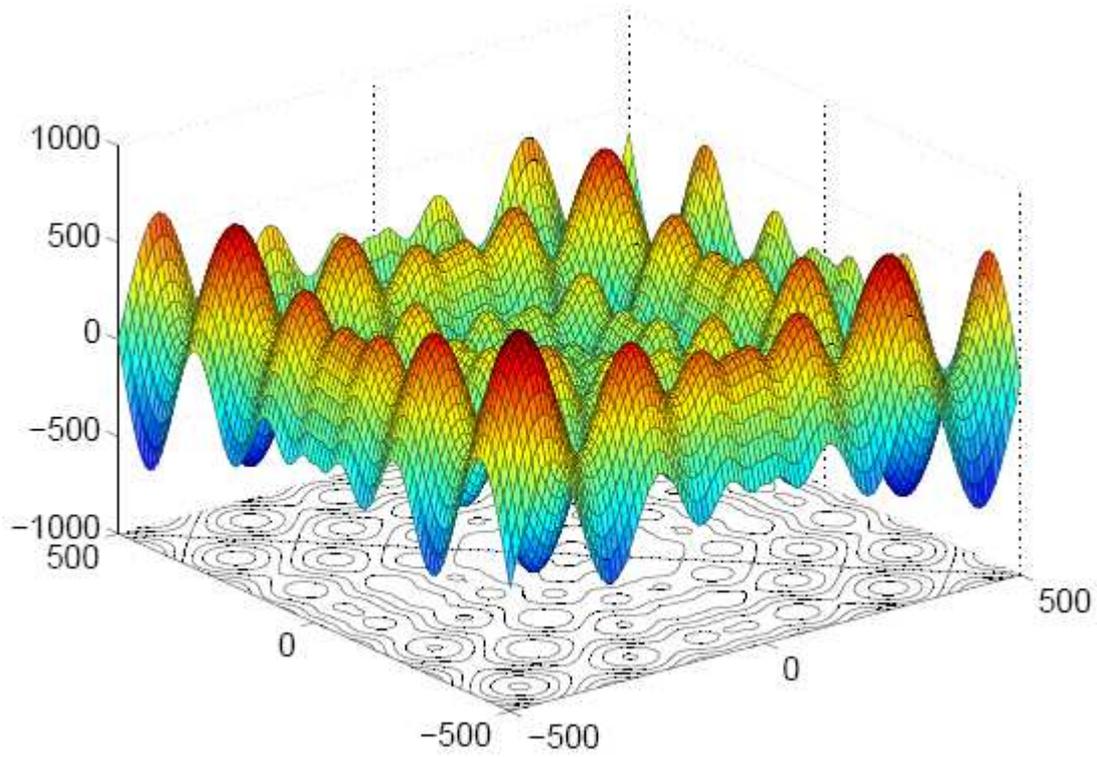
F5 – FUNÇÃO ACKLEY

$$f_{Ackley}(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$$



F6 – FUNÇÃO SHWEFEL

$$f_{Schwefel}(\mathbf{x}) = 418.9829 * N - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|}))$$



Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)