

Multi-Ring: Uma nova topologia para otimização por enxame de partículas (PSO)

Trabalho de Conclusão de Curso

Engenharia da Computação

Marcel Pinheiro Caraciolo
Orientador: Carmelo José Albanez Bastos Filho

Recife, maio de 2008

Multi-Ring: Uma nova topologia para otimização por enxame de partículas (PSO)

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Marcel Pinheiro Caraciolo
Orientador: Carmelo José Albanez Bastos Filho

Recife, maio de 2008

Marcel Pinheiro Caraciolo

**Multi-Ring: Uma nova topologia para
otimização por enxame de partículas
(PSO)**

Resumo

A heurística de otimização por enxame de partículas (*PSO*) tem sido utilizada para resolver diversos tipos de problemas de otimização e busca. Com a aplicação da técnica *PSO* em problemas onde há uma grande dificuldade na busca de soluções adequadas, novas maneiras de solucionar tais problemas são necessários, principalmente em problemas com alta complexidade e alta dimensionalidade. Muitas variações no algoritmo *PSO* padrão foram propostas, visando alterações na equação de atualização de velocidade. Outras propostas visaram modificações na estrutura de comunicação das partículas do enxame. Neste trabalho, uma nova topologia para o *PSO* estruturado em múltiplos anéis é proposta, com o propósito de melhorar o desempenho do algoritmo através da diversidade provida por um mecanismo de rotação de anéis. A busca por melhores soluções sem que o enxame fique estagnado em regiões onde se encontram soluções sub-ótimas é também um dos principais propósitos desta topologia, visando melhorar a capacidade de convergência de exploração do enxame e fornecer maior qualidade à solução obtida pelo *PSO*. Comparações com as topologias já existentes também são realizadas, aonde os resultados das simulações demonstram que a topologia proposta apresenta em diversos casos melhores resultados que as topologias encontradas na literatura como local e global. Por fim, estudos foram realizados, analisando alguns componentes do mecanismo de rotação proposto a fim de verificar sua influência no desempenho do algoritmo *PSO* com a nova topologia.

Abstract

Particle Swarm Optimization (*PSO*) has been used to solve many different types of optimization problems. By applying *PSO* to problems where the feasible solutions are too difficult to be found, new approaches to solve search problems are required, mainly for hyper dimensional spaces. Many variations on the standard *PSO* approach have been proposed, focusing in the velocity update equation. Other approaches changed the communication topology of the swarm. In this work, a novel *PSO* topology based on multiples rings is put forward. The results achieved showed that the diversity provided by the ring rotations improved the simulation results. The search for best solutions without the swarm be stagnant in regions of the search space where low-quality solutions are placed is also one of the main purposes of this topology, aimed at improve the exploration convergence capability of the swarm and provide a better quality to the solution obtained by *PSO*. A comparison with well known topologies was performed. The simulation results have shown that the proposed topology achieves better results than well known topologies such as star and ring. At last, studies were done, analyzing some components of the rotation mechanism in order to verify their influence in the performance of the *PSO* algorithm with the new topology.

Sumário

Índice de Figuras	5
Índice de Tabelas	6
Tabela de Símbolos e Siglas	7
1 Introdução	9
1.1 Objetivos	10
1.2 Estrutura do trabalho	10
2 Fundamentação Teórica	11
2.1 Otimização	11
2.1.1 Otimização local	12
2.1.2 Otimização global	12
2.2 Busca Heurística	13
2.2.1 Otimização Combinatória	14
2.2.2 Heurística	15
2.2.2.1 MetaHeurística	16
3 Otimização por enxame de partículas	18
3.1 O Algoritmo de otimização por enxame de partículas	18
3.1.1 Considerações iniciais	18
3.1.2 Componentes do Algoritmo PSO	21
3.1.3 Variações e melhorias dos componentes do PSO	24
3.1.4 O algoritmo PSO	25
3.2 Topologias e Troca de informação	29
4 Topologia Multi-Ring	31
4.1 Estrutura do Multi-Ring	31
4.2 Comunicação da topologia	33
4.3 Algoritmo PSO com a Topologia Multi-Ring	35

5	Experimentos	37
5.1.1	Simulador <i>PSO Simulation Shell</i>	37
5.1.2	Funções – Teste	38
5.1.3	Parâmetros de simulação	40
5.2	Resultados	42
5.2.1	Resultados pela variação do número de anéis	42
5.2.2	Resultados pela variação do gatilho de disparos de rotação	50
6	Conclusões e Trabalhos Futuros	52
6.1	Contribuições	52
6.2	Conclusão	53
6.3	Trabalhos futuros	55

Índice de Figuras

Figura 1.	Um exemplo de função com o mínimo local x^*A e o mínimo global x^* .	13
Figura 2.	Composição dos vetores que atuam sobre uma partícula de um enxame.	22
Figura 3.	Fluxograma básico para representar o funcionamento do algoritmo <i>PSO</i> .	26
Figura 4.	Esquema do Fluxo do Algoritmo <i>PSO</i> .	27
Figura 5.	Roteiro do Algoritmo do <i>PSO</i> .	28
Figura 6.	(a) Topologia Local e (b) Topologia Global.	29
Figura 7.	Estrutura da Topologia <i>Multi-Ring</i> .	32
Figura 8.	Exemplo de rotação da Topologia <i>Multi-Ring</i> .	34
Figura 9.	Algoritmo <i>PSO – Multi-Ring</i> com a definição do <i>pbest</i> e <i>ringbest</i> .	36
Figura 10.	Algoritmo <i>PSO- Multi-Ring</i> com a etapa de rotação inclusa.	36
Figura 11.	Interface do Simulador <i>PSO</i> com a topologia <i>Multi-Ring</i> .	38
Figura 12.	Funções de teste utilizadas nos experimentos.	39
Figura 13.	Resultados da função <i>Rosenbrock</i> exibindo as curvas de convergência para várias configurações.	43
Figura 14.	Resultados da função <i>Rastrigin</i> exibindo as curvas de convergência para várias configurações.	45
Figura 15.	Resultados da função <i>Griewank</i> exibindo as curvas de convergência para várias configurações.	46
Figura 16.	Resultados da função <i>Ackley</i> exibindo as curvas de convergência para várias configurações.	48
Figura 17.	Resultados da função <i>Schwefel 1.2</i> exibindo as curvas de convergência para as várias configurações.	49

Índice de Tabelas

Tabela 1.	Identificação dos termos do <i>PSO</i>	21
Tabela 2.	Parâmetros da função-teste.....	40
Tabela 3.	Resultados da função <i>Rosenbrock</i> após 5000 e 10000 iterações.	43
Tabela 4.	Resultados da função <i>Rastrigin</i> após 5000 e 10000 iterações.	44
Tabela 5.	Resultados da função <i>Griewank</i> após 5000 e 10000 iterações.....	46
Tabela 6.	Resultados da função <i>Ackley</i> após 5000 e 10000 iterações.....	47
Tabela 7.	Resultados da função <i>Schwefel 1.2</i> após 5000 e 10000 iterações.	49
Tabela 8.	Resultados obtidos pelas funções variando o gatilho de rotação após 10000 iterações.	51

Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

PSO – Particle swarm optimization (otimização por enxame de partículas) .

Swarm - Enxame de partículas.

Fitness – Função de adaptação para avaliação de desempenho.

pbest – Melhor posição encontrada individualmente pela partícula.

gbest – Melhor posição encontrada por todo o enxame.

C_1 e C_2 - Constantes positivas relacionadas ao termo memória e cooperação da partícula i respectivamente.

R_1 e R_2 - Números gerados aleatoriamente com distribuições uniforme no intervalo $[0, 1]$.

\vec{P}_i - Melhor solução encontrada pela partícula i até o presente momento.

\vec{P}_g - Melhor solução encontrada por todo o enxame até o presente momento.

w - Peso de inércia (*inertia weight*).

V_{max} - Velocidade máxima da partícula.

χ - Coeficiente de encolhimento (*constriction factor*).

Region scaling - Sub-região do espaço de busca.

ringbest - Melhor posição encontrada pelo anel.

ℓ_{best} - Mecanismo de vizinhança local.

g_{best} - Mecanismo de vizinhança global.

Tr – Gatilho de disparo de rotação de um anel (*Trigger*).

Agradecimentos

Agradeço inicialmente a Deus, por ter me dado saúde, sabedoria e determinação para realizar este trabalho e de sua constante presença em minha vida.

À minha família, em especial aos meus pais, Anna Kátia e Marco Antônio, que durante todos esses anos me educaram e me apoiaram para que eu chegasse onde eu estou. Além de terem sido essenciais para formação do meu caráter. Ao meu padrasto Manoel que também contribuiu ao repassar aprendizados de educação e disciplina. A todos os meus irmãos e irmãs que sempre estiveram ao meu lado e me fortaleceram com carinho, amor e respeito.

Um agradecimento especial aos meus tios Ana Maria e Roberto e primos Gustavo, Guilherme, Glenda e Robertinho por terem me acolhido de forma tão agradável e sadia que me fez sentir em casa, ainda que minha residência natal não estivesse por perto.

Aos meus amigos que encontrei durante estes anos de caminhada na POLI, o meu muito obrigado. São tantos nomes a serem citados, mas faço uma menção honrosa aos “Discípulos do Messias”, a turma que me acompanhou desde o início. Valeu, vocês são “os” caras!

Aos professores do DSC, todos vocês participaram fundamentalmente na minha formação acadêmica e são exemplos de profissionalismo e ética. Um agradecimento especial a dois professores que me orientaram de forma impecável durante estes anos de universidade. Ao professor Márcio Cornélio, por ter me dado a oportunidade de realizar um trabalho de iniciação científica e também pelo auxílio através de sugestões e conselhos para o desenvolvimento do meu conhecimento técnico e pessoal. E ao professor Carmelo Bastos, que me orientou de forma incisiva e essencial para que eu chegasse até a conclusão do mesmo. Vale ressaltar também a minha gratidão aos colegas Danillo Carvalho e Péricles Miranda, que também me ajudaram em todo o processo de realização deste trabalho.

Por fim a todos os amigos além do âmbito faculdade aqui e na minha terra natal Aracaju e em especial a Andréa Santos, nestes últimos meses, por terem todos contribuído de alguma forma para o que sou hoje. Vocês, sem dúvidas, contribuíram com este trabalho. A todos meu muitíssimo obrigado!

Capítulo 1

Introdução

Problemas compostos de múltiplas variáveis e diversas soluções estão presentes em uma grande variedade de situações reais. Esses problemas normalmente são caracterizados por circunstâncias em que se desejam maximizar ou minimizar uma função numérica de várias variáveis, em um contexto em que podem existir restrições.

Apesar da existência de diversas técnicas clássicas de otimização para solução desses tipos de problema, a sua falta de robustez, aliada à alta complexidade e custo computacional em problemas com alta dimensionalidade, impulsionaram a busca por novas alternativas.

A sofisticação dos recursos computacionais desenvolvidos nos últimos anos possibilitou também avanços nas técnicas de otimização. Estes avanços foram impulsionados com êxito, especialmente, através dos estudos de métodos heurísticos com maior destaque às metaheurísticas que guiam as heurísticas em facilitar a obtenção de soluções ótimas em diferentes espaços de busca, inclusive com vários máximos ou mínimos locais.

Dentre essas metaheurísticas, os algoritmos evolutivos, em especial os baseados em população, têm tido grande sucesso em solucionar problemas de otimização de múltiplas soluções [1],[2],[3]. Um desses algoritmos que tem sido bastante adotado é o algoritmo conhecido como Otimização por Enxame de Partículas (*PSO, particle swarm optimization*) [4],[5].

O algoritmo *PSO* foi proposto inicialmente por J. Kennedy e R. Eberhart [6] e é um técnica inspirada no comportamento social de bandos de pássaros. A busca por alimento e a interação entre os pássaros ao longo do vôo são modeladas como um mecanismo de otimização. No caso, a área sobrevoada é equivalente ao espaço de busca e encontrar o local com maior quantidade de comida corresponde a encontrar a solução ótima.

O algoritmo é baseado em um modelo simplificado da teoria dos enxames (*swarm* experiência do próprio bando para encontrar a melhor região do espaço de busca.

Desde a introdução do *PSO*, diversos pesquisadores propuseram modificações no algoritmo original. Essas modificações, em geral, visam melhorar o desempenho do algoritmo, seja acelerando sua convergência, seja equipando-o para evitar mínimos locais. Uma das possíveis modificações é na topologia de comunicação das partículas, que é o foco deste trabalho.

1.1 Objetivos

Este trabalho propõe-se a apresentar uma nova topologia para o *PSO* baseada em múltiplos anéis com alguns mecanismos próprios específicos que melhoram o desempenho do algoritmo em comparação às topologias usuais adotadas na literatura. Com isso, esta nova topologia foca na solução de problemas de busca e otimização, especialmente na solução de problemas que sofrem com estagnação, isto é, a apresentação de soluções sub-ótimas. E também visa apresentar mecanismos que possam ser utilizados em outras topologias do algoritmo *PSO*.

1.2 Estrutura do trabalho

Esta monografia está organizada em capítulos. A seguir, está detalhada a estrutura deste trabalho.

O capítulo 2 discute a fundamentação teórica necessária para o entendimento do trabalho. São detalhados os conceitos básicos de otimização e heurísticas de otimização necessários para a compreensão da técnica abordada neste trabalho. O capítulo 3 detalha os conceitos relacionados à heurística de otimização por enxame de partículas. Isso inclui a apresentação do algoritmo e a estrutura do mesmo. Dentre os componentes estruturais apresentados, destaca-se a topologia de comunicação entre as partículas, que é objetivo-alvo deste trabalho.

O capítulo 4 introduz a topologia proposta, *Multi-Ring*. Serão detalhados a estrutura e os mecanismos específicos envolvidos nesta topologia voltada para a heurística de otimização por enxame de partículas. O capítulo 5 detalha os experimentos realizados com o *PSO* modificado com a nova topologia, relatando os problemas teste escolhidos para a elaboração dos experimentos, bem como a análise dos resultados obtidos experimentalmente.

E, por fim, o último capítulo conclui este trabalho, detalhando as principais contribuições referentes a este trabalho e indicando possíveis trabalhos futuros. Um apêndice com as publica

Capítulo 2

Fundamentação Teórica

Neste capítulo são descritos os conceitos básicos de otimização e heurísticas de otimização necessários para a compreensão da metodologia usada na realização deste projeto.

2.1 Otimização

Tornou-se natural ao ser humano propor formulações matemáticas para encontrar a solução de muitos problemas encontrados na natureza. A teoria da otimização pode ser definida como sendo a tarefa de determinar qual das soluções existentes para um determinado problema é a melhor, isto é, a ótima. Esta determinação é feita através de métodos que, em grande parte dos casos, são formulados formalmente através da matemática.

O modelo matemático básico de um problema de otimização obedece ao seguinte raciocínio: minimizar uma função-objetivo $f(x)$, de acordo com as restrições impostas pela função $g(s)$, onde s representa o conjunto de soluções permitidas para a função em questão e x é variável que deve ser otimizada.

Alguns destes problemas envolvem apenas modelos lineares, nos quais as variáveis são contínuas e apresentam comportamento linear, tanto em relação à função objetivo quanto às restrições [7]. Os outros problemas são conhecidos como problemas de otimização não-lineares, cujas características são as de exibir qualquer tipo de não-linearidade ou não continuidade, seja na função objetivo ou em qualquer das suas restrições. Naturalmente, eles apresentam a maior dificuldade e complexidade nas suas resoluções e são o foco deste trabalho [7].

Define-se que a solução ótima encontrada pode ser de natureza global ou local. É dita global a melhor solução dentre todas as soluções existentes para aquele determinado problema.

Entretanto, quando a solução encontrada é a melhor em um conjunto restrito de soluções que se nas próximas seções.

2.1.1 Otimização local

Pode-se definir um ponto mínimo local x_A^* de uma função $f(x)$, para uma determinada região A , da seguinte forma:

$$f(x_A^*) \leq f(x), \forall x \in A, \quad (1)$$

onde $A \subset S \subseteq R^n$ e S denota o espaço de busca. Vale destacar que $S = R^n$ em problema sem restrições e também que A é um subconjunto de S . O espaço de busca S pode conter múltiplas sub-regiões A_i tal que $A_i \cap A_j = \emptyset$ quando $i \neq j$. Sendo que pontos definidos nessas subregiões são considerados únicos ($x_{A_i}^* \neq x_{A_j}^*$). Qualquer $x_{A_i}^*$ pode ser considerado como o mínimo de A_i . Não há restrições no valor que a função pode assumir para o mínimo, logo $f(x_{A_i}^*) = f(x_{A_j}^*)$ é permitido. Com isso, o mínimo é definido por $x_{A_i}^*$.

A grande parte dos algoritmos de otimização exige um ponto de partida $x_o \in S$. Um algoritmo de otimização local eficaz deve garantir que será capaz de encontrar um mínimo local $x_{A_i}^*$ para um conjunto A , se $x_o \in A$.

2.1.2 Otimização global

O *mínimo global*, x^* de uma função $f(x)$, pode ser definido da seguinte forma:

$$f(x^*) \leq f(x), \forall x \in S, \quad (2)$$

onde S denota o espaço de busca. Assim como a otimização local descrito acima, a otimização global também começa pela escolha de um ponto inicial $x_o \in S$ e para problemas sem restrições é comum usar $S = R^n$, onde n é o número de dimensões de x .

Dennis e Schnabel [8] definem com outra visão os algoritmos de otimização global. Eles os definem como sendo algoritmos capazes de localizar o mínimo (local) de $A \subset S$, independentemente da posição atual de x_o . Esses algoritmos consistem em duas etapas: a fase dos “passos globais” e a fase dos “passos locais”. Os passos globais são projetados para garantir que o algoritmo passará para região A_i , e o processo dos passos locais será responsável por encontrar o mínimo de A_i . Isto significa que eles podem convergir a um mínimo local independente de sua posição inicial x_o . Estes métodos são chamados de algoritmos globalmente convergentes, e

podem ser usados tanto para encontrar mínimos locais como globais. Entretanto, no caso de mínimos globais não há nenhum modo geral seguro e conhecido para garantir a sua localização.

A Figura 1 ilustra a diferença entre um mínimo local e o mínimo global.

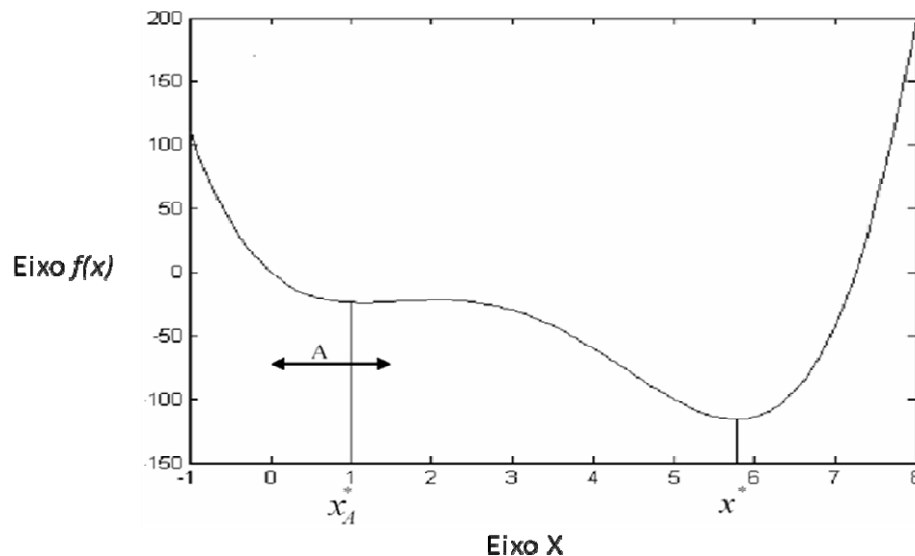


Figura 1. Um exemplo de função com o mínimo local x_A^* e o mínimo global x^* .

2.2 Busca Heurística

O processo de busca e otimização possui vários componentes, como o espaço amostral de busca, onde são consideradas todas as possibilidades de solução de um determinado problema, e a função de custo (função-objetivo), que define uma forma de avaliar os elementos dentro do espaço de busca.

Existem muitos métodos de busca e funções de custo. Grande parte das técnicas de busca e otimização tradicionais são iniciadas com um único candidato, que iterativamente é manipulado utilizando métodos específicos diretamente associados ao problema a ser solucionado. Geralmente, estes métodos são monotônicos devido à manipulação de um único indivíduo, e sua simulação em computadores pode ser muito complexa. Apesar disto, na prática eles são bastante utilizados com sucesso em diversas aplicações. Pode-se citar como exemplo, os algoritmos de otimização por *simulated annealing* que usam a simulação do processo de recozimento (*annealing*) da metalurgia para a solução de problemas [24].

Alternativamente, existem as técnicas de otimização e busca combinatórias, que operam diferentes áreas do espaço de solução possíveis, combinando um número de elementos apropriados em várias regiões.

Os itens a seguir tratarão dos conceitos relacionados às técnicas de otimização combinatória e suas ramificações, pois são estas que fazem parte do escopo deste trabalho.

2.2.1 Otimização Combinatória

Conforme descrito nas seções 2.1.1 e 2.1.2, a solução ótima encontrada pode ser de aspecto global ou local. Quando é possível atingir o ótimo global de um problema com a utilização de tempo e recursos viáveis, por meio de metodologias matemáticas, classifica-se este problema como um problema de resolução matemática. Entretanto, quando para atingir o possível ótimo global a resolução está associada a um alto custo computacional, este problema é classificado como problema de otimização combinatória.

A otimização combinatória é caracterizada matematicamente como um processo de localização de um subconjunto de elementos de um espaço de busca, em que cada um destes elementos associados a um peso, apresente um somatório ótimo. Através da maximização ou minimização da função-objetivo, os elementos localizados que compõem esse subconjunto serão responsáveis pela solução ótima para o problema em questão [7].

Todavia, os problemas classificados como de otimização combinatória podem possuir uma ampla quantidade de soluções. Dessa forma, torna-se altamente custoso buscar a solução ótima, visto que localizar o ótimo global exige percorrer todo o espaço de busca, especialmente se o problema abordado for de alta dimensionalidade. Portanto, problemas dessa natureza necessitarão de um excessivo custo computacional e um longo tempo de processamento.

Esses problemas estão presentes em uma grande variedade de tarefas de otimização. Situações que envolvem a escolha de uma melhor trajetória a ser seguida por um veículo ou a solução de um problema matemático complexo e não contínuo fazem parte deste grupo [7].

Diante de tal fato, pesquisadores, buscando avanços nesta área, decidiram unir aos algoritmos implementados para solucionar problemas de otimização combinatória técnicas de computação inteligente. A conclusão óbvia foi que a introdução de alguma “inteligência” poderia reduzir o custo computacional e tornar viável o tempo para alcançar do ótimo global.

A sofisticação dos recursos computacionais desenvolvidos nos últimos anos tem motivado

Priorizou-se então por buscar a solução “boa” de um problema, pois estas podem ser perfeitamente aceitáveis dependendo do tipo do problema. São soluções “boas” ou sub-ótimas, aquelas que se destacam por atender algum pré-requisito definido no início da busca.

O uso de técnicas de computação inteligente associadas à otimização combinatória apresentou resultados com grande êxito [9]. Não somente reduzindo o custo computacional associado a esses tipos de problemas como também permitindo a solução de problemas com alta dimensionalidade. Pode-se então definir uma heurística como sendo uma técnica de busca de soluções que encontra uma solução aceitável para um problema com um custo computacional razoável. Vale considerar que a solução encontrada pela heurística não é exatamente o ótimo global do problema e não contém informações de quão próximo a solução encontrada está dele.

2.2.2 Heurística

No âmbito computacional, as heurísticas são definidas como uma tarefa que tem por objetivo encontrar a solução mais aproximada através de algum conhecimento disponível e com médio esforço computacional [7]. Uma definição dada por Reeves [10] é que uma heurística é uma técnica que busca boas soluções (quase ótimas) a um custo computacional razoável, porém sem ser capaz de garantir que as mesmas sejam ótimas ou admissíveis. Em alguns casos, pode-se nem determinar quão próximo do ótimo se encontra a solução admissível.

Logo, a heurística não garante descobrir a melhor solução, isto é, a solução ótima para um problema. O seu principal foco está na flexibilidade e na busca contínua de utilização mínima de recursos computacionais. Dessa forma, é possível que se atinja o ótimo global para um determinado problema, mesmo que a heurística não garanta esta responsabilidade e nem indique quão próxima a solução se encontra do ótimo global.

Os algoritmos baseados em heurísticas, em sua maioria, possuem um caráter genérico [11]. Como se torna bastante dispendioso desenvolver uma heurística específica para cada problema existente, visto que, exigiria a presença de um especialista no problema para sua implementação, e conseqüentemente, um considerável tempo para a sua implementação, pesquisadores tomaram por decisão tentar criar heurísticas de caráter genérico [11].

Apesar de não se ter encontrado a heurística que encontra o ótimo global a um baixo computacional, os estudos nesta área ainda não foram esgotados. Diversas heurísticas com ótimos

resultados são constantemente propostas. Basicamente, agrupam-se em quatro grupos: heurísticas

Destas, a heurística que se enquadra no escopo deste trabalho é a metaheurística e será ela a detalhada no item a seguir.

2.2.2.1 MetaHeurística

Assim como as demais heurísticas, a metaheurística visa também encontrar uma solução aceitável para um problema sem fazer uso de rigidez ou elevada quantidade de recursos computacionais. Seu foco principal é encontrar a resposta a que se propõe utilizando um razoável grau de recursos computacionais e flexibilidade controlada.

Entretanto, as metaheurísticas se diferenciam das outras heurísticas, pois seus algoritmos são aplicáveis a vários tipos de problema fazendo uso de combinação de uma ou mais heurísticas para explorarem de forma conjunta o espaço de soluções. Para isso, na maioria dos casos, eles se inspiram em diversas áreas como física, biologia, ciências sociais e entre outros.

Uma outra característica vantajosa das metaheurísticas é a facilidade para a exploração em espaços de busca dos ótimos locais. Enquanto a heurística seleciona um ótimo local como o melhor, já que não encontrou outro, dentro de um espaço de busca determinado por uma função, as metaheurísticas fazem uso de táticas que permitem a melhor exploração deste espaço, e conseqüentemente, permitir facilitar a localização de um melhor ótimo local [13].

Diversas metaheurísticas foram propostas. As principais características que diferenciam as diferentes abordagens são [13]:

- A definição da escolha e representação da solução inicial;
- A definição da vizinhança $N(t)$ de uma solução t ;
- O critério de seleção de uma solução vizinha dentro de $N(t)$;
- A condição de término.

Diante destas características, agrupou-se as metaheurísticas em dois grandes grupos de acordo com os critérios utilizados para a busca de soluções: busca local e busca populacional.

O algoritmo das metaheurísticas do grupo de busca local se baseia na exploração do espaço de soluções através de movimentos entre seus vizinhos. A cada iteração uma nova geração é gerada até que se consiga atingir uma solução desejável. Exemplo de um algoritmo qualificado como metaheurística de busca local são os algoritmos de busca tabu [13] e *simulated annealing* [24]. Estes

algoritmos determinam a cada iteração, através de uma função de avaliação, qual vizinho conseguiu

Diferente das metaheurísticas caracterizadas pela busca local, os algoritmos das metaheurísticas de busca populacional consistem na busca da manutenção de um conjunto de boas soluções e, através de combinações entre elas, tentam alcançar soluções melhores. Como exemplo desta classe de metaheurísticas, pode-se citar os algoritmos genéticos, visto que a geração de uma nova população (possíveis soluções) é criada a partir de operações de recombinação e mutação entre as soluções [9]. Logo, a evolução da população faz com que a formação dos novos indivíduos caminhe para o ótimo, à medida que aumenta sua função de adaptação (*fitness*).

Dentre essas metaheurísticas, várias propostas de novos procedimentos metaheurísticos vêm sendo apresentadas. Uma dessas propostas, visando chegar cada vez mais próximo da solução ótima em tempo cada vez menor e que tem sido bastante adotado, é o algoritmo conhecido como otimização por enxame de partículas (*Particle Swarm Optimization*) [6]. Esse algoritmo faz parte das mais recentes metaheurísticas propostas e está, até o presente momento, apresentando excelentes resultados para problemas de alta dimensionalidade.

Capítulo 3

Otimização por enxame de partículas

Este capítulo visa apresentar e explicar os conceitos básicos da heurística de otimização por enxame de partículas (*PSO*) relevantes para o escopo deste trabalho. São detalhados os componentes principais e o comportamento do algoritmo.

3.1 O Algoritmo de otimização por enxame de partículas

3.1.1 Considerações iniciais

A técnica de otimização por enxame de partículas (*PSO*) é uma metaheurística introduzida por James Kennedy e Russel Eberhart na década de 90 e emergiu a partir de experiências com algoritmos que modelam o “comportamento social” observado em algumas espécies de pássaros [4],[6].

Esta técnica é inspirada na natureza e por isso é considerado por alguns como uma técnica de computação evolutiva. As técnicas de computação evolutiva são baseadas em mecanismos biológicos, que visam solucionar problemas de otimização combinatória implementados em computador.

Uma das características que identifica um algoritmo evolutivo é a existência de uma população de soluções. Nela estará presente a solução resultado para o problema em questão. Para avaliar o desempenho dessa população, faz-se necessário o uso de uma função de adaptação que irá atribuir “notas” de acordo com a aptidão, isto é, o grau de proximidade em que a solução se encontra da solução ótima. Através de mecanismos que provoquem alterações nesta população,

objetivando a formação de novas soluções, a idéia é que a população evolua em direção ao ótimo,

A definição de enxame pode ser considerada como um conjunto de indivíduos que interagem localmente entre si, regidos por um comportamento global, buscando a solução para problemas de forma distribuída [14].

A interação entre os membros do enxame pode ser de forma direta, através dos seus vizinhos mais próximos, ou de forma indireta, através da troca de informações do comportamento dos seus vizinhos mais distantes. Normalmente, cada elemento do enxame faz uso da sua própria experiência e da experiência do próprio enxame. Isso ressalta o senso do social, onde os indivíduos aprendem com o sucesso dos seus vizinhos e sempre priorizam o objetivo do grupo. Desta forma, não há centralização de controle e o grupo consegue atingir as soluções desejadas de forma igualitária.

Um enxame possui inteligência se certas características estiverem presentes [14]. Essas características são descritas por princípios básicos:

- Princípio da proximidade: Afirma que os membros do enxame devem ter a noção do espaço e tempo durante seu deslocamento de modo que mudanças possam ser realizadas sem interferir na movimentação de seus vizinhos.
- Princípio da qualidade: Afirma que os membros do enxame devem ser capazes de se adaptar aos fatores de qualidade do ambiente, isto é, a depender do objetivo almejado, a população deve buscar um caminho em que resulta na melhor solução, garantindo qualidade para o seu bando.
- Princípio da reposta diversa: Afirma que o enxame não deve alocar todos seus recursos em um só canal. As mudanças no processo são sempre desejadas a fim de que o enxame possa evoluir em sua busca. Isso previne que os membros fiquem presos em um local em que não se encontra a melhor solução (mínimo local). Porém, é desejável que haja ações corretivas caso haja mudanças repentinas, pois do contrário podem ocorrer desvios no processo evolutivo que podem acarretar em um aumento no esforço computacional.
- Princípio da estabilidade: Diz que os membros do enxame não devem mudar o seu comportamento a cada mudança do ambiente. O enxame deve ser capaz de avaliar

se tal mudança é necessária a fim de aplicar um esforço conjunto de todo o bando.

com seu objetivo.

- Princípio da adaptabilidade: Afirma que a população deve mudar seu comportamento quando o custo computacional envolvido valer a pena, isto é, o enxame deve possuir a capacidade de avaliar as mudanças ocorridas e decidir se é válida a mudança no comportamento do bando.

A existência dessas características é extremamente importante para garantir o equilíbrio e a estabilidade do bando durante a evolução do processo, qualificando-o, assim, como inteligente.

Há anos vem sendo estudado o comportamento social de alguns grupos de animais como a organização das abelhas, formigas e pássaros na busca de alimentos ou novos locais para estabelecer sua nova moradia. Este último grupo, em especial despertou um grande interesse de alguns pesquisadores, na década de 80, onde se destaca dentre eles o biólogo Frank Heppner [15].

Após diversas observações sobre o comportamento de bando de pássaros em revoada, Heppner decidiu por modelar aquela inteligência coletiva para usá-la em métodos de busca para solução de problemas. Os estudos de Heppner consideravam que o comportamento de várias espécies de pássaros, em bando ao longo do seu vôo, fazia o uso de alguma lógica e de alguma forma de comunicação. Após vários estudos e observações, Heppner descreveu o raciocínio por trás daquele comportamento, qualificando-o como comportamento social [15].

James Kennedy e Russel Eberhart, inspirados no comportamento social dos pássaros estudados por Heppner, desenvolveram uma técnica de otimização que veio a ser conhecida como enxame de partículas. Essa denominação se deu, pois se notou que o modelo descrito por Heppner demonstrava características de um enxame inteligente, onde seus membros que apresentavam tal comportamento foram generalizados para o termo partículas.

Não só o nome do algoritmo, como os demais aspectos do modelo estudado por Heppner ganharam uma nova conotação. A busca pelo alimento ou pelo ninho e a interação entre os pássaros ao longo do vôo são modelados como um mecanismo de otimização.

Fazendo uma analogia, o termo partícula foi adotado para simbolizar os pássaros e representar as possíveis soluções do problema a ser resolvido. A área sobrevoada pelos pássaros é equivalente ao espaço de busca e encontrar o local com comida, ou o ninho, corresponde a encontrar a solução ótima. Para que o bando de pássaros sempre se aproxime do objetivo, ao

invés de se perder ou nunca alcançar o alvo focado, utiliza-se o indicador denominado *fitness*,

alimentos ou os ninhos, os pássaros fazem uso de suas experiências e da experiência do próprio bando. O termo indicador da experiência ou conhecimento individual de cada partícula, isto é, seu histórico de vida, é o *pbest*. Em uma abordagem mais simples, o responsável por representar o conhecimento do enxame como um todo é o *gbest*. A Tabela 1 apresenta de forma resumida as nomenclaturas descritas acima:

Tabela 1. Identificação dos termos do *PSO*.

Termo	Significado
Partícula	Pássaro
Enxame	Bando de pássaros
Espaço de Busca	Área sobrevoada pelos pássaros
Posição	Localização de cada pássaro durante o voo
Solução ótima	Localização do pássaro onde ele encontrou o alimento ou o ninho
Fitness	Função de avaliação
<i>pbest</i>	Melhor posição conhecida pelo pássaro (Experiência individual)
<i>gbest</i>	Melhor posição conhecida pelo enxame (Experiência coletiva)

As outras particularidades referentes ao enxame de partículas são representadas por formulação matemática e serão detalhados na seção a seguir.

3.1.2 Componentes do Algoritmo PSO

O algoritmo da técnica de otimização por enxame de partículas é um algoritmo que possui uma população de partículas, onde cada partícula representa uma possível solução para o problema de otimização. Cada partícula \vec{x}_i do enxame pode ser representada por um objeto que sofre ação de influência por três forças representadas matematicamente em forma de vetor:

inércia, memória e cooperação. O vetor inércia representa a força que impulsiona a partícula a seguir a direção a que tendia previamente. O vetor memória faz com que a partícula se direcione de acordo com seu histórico, ou seja, baseado nas suas experiências individuais passadas, ela se move aos lugares que se mostraram favoráveis ao problema em questão. Por último, o vetor cooperação que direciona a partícula para a melhor direção já conhecida pelo enxame, ou seja, utiliza o histórico baseado nas experiências coletivas de todo o enxame [16].

O deslocamento de cada partícula é representado pelo vetor resultante da composição da influência do conhecimento da partícula (vetor memória), associado ao conhecimento do bando (vetor cooperação), mais a trajetória em que a partícula vinha adotando (vetor inércia). A Figura 2 ilustra a atuação desses vetores sobre uma partícula representada pelo ponto em um tempo t .

Figura 2. Composição dos vetores que atuam sobre uma partícula de um enxame.

Como se pode ver na Figura 2, o ponto representa a nova posição da partícula no espaço. A nova posição da partícula é resultado da adição do vetor posição atual com o novo vetor de velocidade a ser adotada. A nova posição das partículas pode ser calculada pela seguinte equação[16]:

$$, \quad (3)$$

onde:

é novo vetor posição de cada partícula i no tempo $t + 1$;

é o vetor posição atual de cada partícula i no tempo t ;

é o vetor velocidade dessa partícula;

Δt equivale ao espaço de tempo considerado.

O vetor velocidade é um dos itens mais importantes no algoritmo *PSO*, pois as mudanças de velocidade. As partículas voam pelo espaço de busca tendo suas velocidades atualizadas dinamicamente de acordo com o histórico das experiências individuais e coletiva de todo o enxame [17]. Logo, a evolução do algoritmo do *PSO* está associada à trajetória percorrida pelo enxame e ao tempo gasto para encontrar a melhor solução do problema. O vetor velocidade é representado pela seguinte equação [16]:

$$\vec{V}_i(t+1) = \vec{V}_i(t) + C_1 R_1 (\vec{P}_i - \vec{X}_i(t)) + C_2 R_2 (\vec{P}_g - \vec{X}_i(t)), \quad (4)$$

Onde:

$\vec{V}_i(t+1)$ é o novo vetor velocidade da partícula i ;

$\vec{V}_i(t)$ é o vetor velocidade atual da partícula i ;

C_1 e C_2 são uma constante positiva relacionada ao termo memória e cooperação da partícula i respectivamente;

R_1 e R_2 são números gerados aleatoriamente com distribuições uniforme no intervalo $[0,1]$.

\vec{P}_i é a melhor solução encontrada pela partícula i até o presente momento;

\vec{P}_g é a melhor solução encontrada por todo o enxame até o presente momento;

$\vec{X}_i(t)$ é a posição atual da partícula i .

As constantes C_1 e C_2 , também chamadas na literatura de parâmetros de confiança, aplicadas tanto ao vetor memória quanto ao vetor cooperação indicam a importância do caráter social ou cognitivo do enxame. Altos valores atribuídos a C_1 , que está associado ao vetor memória, indica que o algoritmo dará mais importância ao conhecimento individual da partícula. Por outro lado, altos valores atribuídos a C_2 , que está associado ao vetor cooperação, indica que o conhecimento global do enxame tem maior importância no algoritmo. Na literatura, tem se adotado para a resolução de grande parte dos problemas, valores semelhantes aos C_1 e C_2 , onde a atualização da velocidade sofrerá influência de forma equivalente tanto do conhecimento individual da partícula como o conhecimento do enxame [17].

As variáveis \vec{P}_i e \vec{P}_g representam as melhores soluções encontradas pela partícula individualmente e socialmente respectivamente. Eles distinguem os dois componentes do termo de atualização da velocidade do *PSO*: o componente de cognição e o componente social. A

interação social entre as partículas é extremamente importante e desenvolve a cooperação entre

3.1.3 Variações e melhorias dos componentes do PSO

Novas melhorias foram propostas para os componentes do algoritmo de otimização por enxame de partículas. Essas melhorias visam melhorar a velocidade de convergência do *PSO*, ou seja, envolvem mudanças na equação da atualização de velocidade sem mudar na estrutura do próprio algoritmo. Isso resulta em otimização local de melhor desempenho a depender do problema a ser tratado [18].

A introdução do fator inércia (*inertia weight*) foi uma das primeiras modificações no algoritmo PSO original objetivando melhorar a velocidade de convergência, ou seja, a rapidez com que as partículas encontram a solução ótima. A inércia é um fator escalar e está associado à atualização da velocidade, podendo também ser dinamicamente modificada a cada iteração. A junção da equação de atualização de velocidade com o fator inércia resulta na seguinte equação[18]:

$$\vec{V}_i(t+1) = w\vec{V}_i(t) + C_1 R_1 (\vec{P}_i - \vec{X}_i(t)) + C_2 R_2 (\vec{P}_g - \vec{X}_i(t)), \quad (5)$$

onde a equação é a mesma da equação de atualização proposta anteriormente, diferindo apenas na adição de um peso de inércia w no ajuste da velocidade.

Diversos estudos foram realizados sobre os efeitos dos valores de w sobre a atualização de velocidade. Os resultados obtidos com a variação da inércia com o passar do tempo demonstram uma convergência mais rápida do que os mesmos obtidos com valores de inércia estáticos. Sugere-se que durante o processo de otimização, w iniciando com valores altos próximos a 1,0 encoraja as partículas a uma exploração maior do espaço de busca (exploração de amplitude) e eventualmente esses valores vão decrescendo (abaixo de 1,0), focando assim os esforços do enxame na melhor área encontrada durante a exploração (exploração de profundidade) [20]. A inclusão do peso de inércia resultou em uma melhora de desempenho e extinguiu a necessidade da adição de um mecanismo de controle de limite de velocidade (V_{max}) na própria equação de velocidade.

Outra melhoria proposta para ajudar na convergência foi a introdução do coeficiente ou fator de encolhimento (*constriction factor*) [19],[20]. O modelo do coeficiente de encolhimento propõe um novo método de escolha dos valores de w , C_1 e C_2 de forma que a convergência seja assegurada. Ele descreve que estes parâmetros passem a ser determinados analiticamente através

de coeficiente de encolhimento χ . A equação de atualização de velocidade usando este coeficiente

$$\vec{V}_i(t+1) = \chi [\vec{V}_i(t) + C_1 R_1 (\vec{P}_i - \vec{X}_i(t)) + C_2 R_2 (\vec{P}_g - \vec{X}_i(t))], \quad (6)$$

Onde:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \varphi = C_1 + C_2 \text{ e } \varphi > 4. \quad (7)$$

Análises matemáticas demonstram que quando utilizado a constante (7) $\varphi > 4$, a convergência até a melhor solução encontrada ocorre de forma rápida e garantida. Os parâmetros C_1 e C_2 em maior parte das implementações com a introdução do fator de encolhimento utilizam valores semelhantes com a finalidade de simplificar o processo. Usando a constante $\chi = 4,01$, demonstrou-se que a convergência é segura, obtendo $C_1 = 2,05$ e $C_2 = 2,05$. Então, este fator é aplicado na equação de velocidade, implicando uma redução de velocidade a cada iteração, e conseqüentemente na melhora de desempenho em comparação aos algoritmos com fator de inércia e padrão [19].

Entretanto, ambas as propostas de modificação na equação de velocidade detalhadas nessa seção visam o balanceamento da habilidade de exploração de amplitude e profundidade, que têm em vista melhorar a velocidade de convergência e a qualidade da solução encontrada. Neste trabalho, focou-se no algoritmo *PSO* com o fator de encolhimento, já que o mesmo tem sido utilizado em diversas publicações para atingir a solução ótima em problemas de otimização com funções não-lineares multimodais.

3.1.4 O algoritmo PSO

O algoritmo básico de otimização por enxame de partículas pode ser descrito brevemente utilizando os seguintes passos: dada uma população inicial de partículas, atualiza-se o vetor posição a partir do vetor velocidade de cada partícula até que se atinja o critério de parada pré-definido. A Figura 3 ilustra essa lógica [6]:

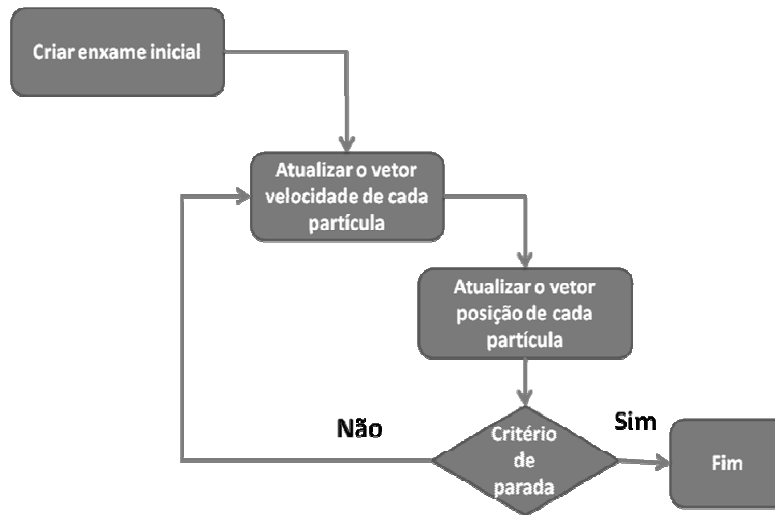


Figura 3. Fluxograma básico para representar o funcionamento do algoritmo *PSO*.

O enxame é inicializado com os valores dos vetores de velocidade e posição gerados aleatoriamente. Para que não haja descontrole, isto é, uma explosão na velocidade, define-se um valor limite para estas velocidades. Assim, a velocidade de cada partícula deve estar sempre entre [17]. Este mecanismo, também conhecido como fronteira, restringe a velocidade da partícula, caso ela ultrapasse esta fronteira. Sendo assim, caso a velocidade de alguma partícula ultrapasse os limites desse intervalo, ela será imediatamente atualizada para o valor do limite mínimo ou máximo de velocidade.

Da mesma maneira, definem-se valores limites para as posições do espaço de busca [17]. A depender do problema de otimização a ser tratado, sugerem-se valores para fronteiras do espaço de busca sobre a qual as partículas devem ser inicializadas. Assim, caso a posição da partícula ultrapasse as fronteiras do espaço de busca, como no caso da velocidade, ela será imediatamente atualizada para o valor do limite máximo ou mínimo da fronteira.

Um mecanismo também utilizado na inicialização das posições das partículas do enxame no início do algoritmo é o uso da partição do espaço de busca (*region scaling*) [17]. Este mecanismo tem como objetivo inicializar a posição de cada partícula nos limites de uma sub-área do espaço de busca onde sobre a mesma não tenha a posição que tenha a solução ótima. Isso elimina a possibilidade de o enxame alcançar a solução ótima do problema sem ter explorado todo o espaço de busca. Este método é bastante aplicado em pesquisas de novos algoritmos *PSO* para fins de análise de comparação e testes de desempenho onde a solução ótima do problema já é conhecida.

A primeira iteração do algoritmo inicia com a atribuição de valores aos parâmetros da equação de velocidade. Definem-se então os valores referentes ao enxame, constantes e o critério de parada. Tendo já definido os valores para posição das partículas e suas respectivas velocidades, aplica-se o cálculo do *fitness* a cada partícula desta população. Conforme explicado na seção 3.1.1, o *fitness* avalia o desempenho da partícula.

Com as partículas do enxame avaliadas, extraem-se os *pbest* e o *gbest*, isto é, a melhor posição encontrada pela partícula e pelo enxame. Depois as velocidades e as posições de cada partícula do enxame são atualizadas. Diante das novas posições, caso o critério de parada tenha sido atingido, a solução do problema encontrada é apresentada. Caso contrário, aplica-se novamente o *fitness* a este enxame, atualizam-se os valores de *pbest* e *gbest*, caso seja apresentada uma solução melhor, seguido da velocidade e posição de cada partícula do enxame. O laço prossegue até o critério de parada ter sido atingido. A Figura 4 ilustra o esquema do algoritmo com mais detalhes [17].

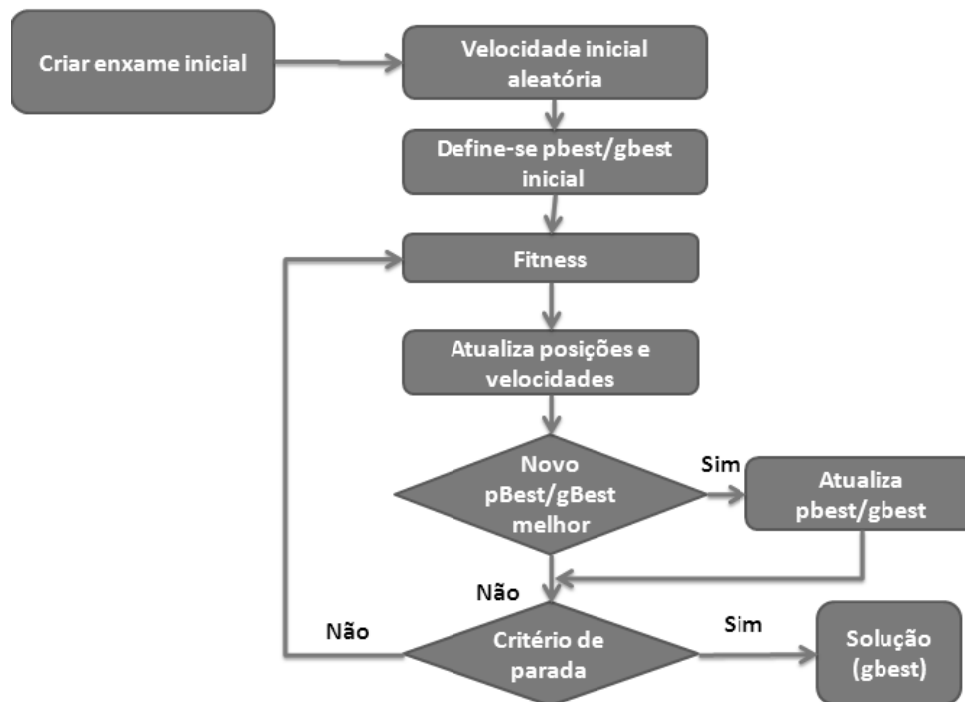


Figura 4. Esquema do Fluxo do Algoritmo *PSO*.

Pode-se perceber pelo esquema acima apresentado que o algoritmo *PSO* é bastante

Roteiro passo a passo do algoritmo básico PSO

Passo 1: Inicializa-se aleatoriamente as posições (\vec{x}_i) e velocidade (\vec{v}_i) do enxame;

Passo 2: Atribui-se *pbest* ($\vec{P}_i(t)$) para as posições atuais das partículas;

Passo 3: Atribui-se *gbest* ($\vec{P}_g(t)$) a melhor posição encontrada pelo enxame;

Passo 4: Aplica-se a avaliação do Fitness $f(\vec{x}_i)$ ao enxame;

Passo 5: Atualiza-se os valores de velocidade e posição utilizando as equações de velocidade e posição (3),(4);

Passo 6: Se o novo *pbest* e *gbest* possuem melhores valores, atualizam-se tais parâmetros;

Passo 7: Caso o critério de parada tenha sido alcançado, apresenta-se a solução (*gbest*). Caso o contrário, incrementa-se o tempo e retorna ao passo 4.

Figura 5. Roteiro do Algoritmo do *PSO*.

Existem duas variantes para a escolha do critério de parada do algoritmo *PSO*. Uma é pelo número de iterações, ou seja, quando o algoritmo chega ao fim por que atingiu a última iteração. A outra é pela função de avaliação (*Fitness*), ou seja, quando o algoritmo chegou ao fim por que alcançou um valor pré-definido para a função. Para este trabalho, foi adotado o número de iterações como critério de parada, onde o mesmo foi utilizado em vários exemplos em que se aplicou *PSO* e se obteve sucesso [15],[17],[18].

Além das modificações na equação da velocidade propostas, outro componente importante que influencia no desempenho do algoritmo *PSO* também tem sido tema de estudos. Este componente é a estrutura ou topologia de comunicação das partículas e rege como as partículas do enxame trocam informações. Na próxima seção, serão apresentados os conceitos relacionados à topologia e troca de informações entre as partículas.

3.2 Topologias e Troca de informação

A troca e disseminação de informação entre as partículas do enxame é o principal componente de qualquer enxame inteligente baseado no algoritmo de otimização por enxame de partículas. O *PSO*, assim com os outros algoritmos de otimização por enxame tem mecanismos próprios para disseminar as melhores posições encontradas durante a execução do algoritmo [21],[22]. Um mecanismo utilizado pelo enxame para distribuir essas informações é denominado de topologia. Esta estrutura rege a organização das partículas no enxame e como as mesmas se comunicam. Logo, variações sobre essa estrutura podem influenciar no desempenho do algoritmo.

A escolha da topologia influencia na avaliação da velocidade das partículas. A depender de como as partículas se comunicam entre si e do problema a ser tratado, a busca pela solução ótima pode priorizar tanto a velocidade de convergência, a qualidade da solução ou ambas. As principais topologias utilizadas como mecanismos de comunicação entre as partículas são: a topologia global e a topologia local. A Figura 6 apresenta a estrutura de tais topologias.

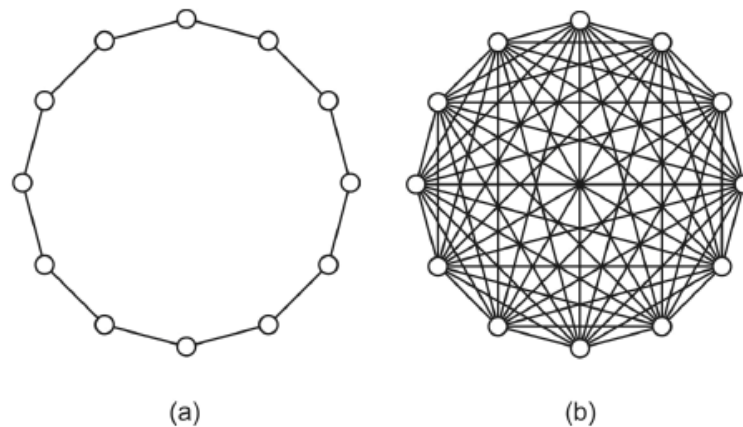



Figura 6. (a) Topologia Local e (b) Topologia Global.

Na topologia local, conforme ilustrado na Figura 6.a, o enxame está organizado em formato de anel e cada partícula deste possui dois vizinhos. Logo, a partícula troca informações apenas com seus vizinhos diretos.

Esse mecanismo de troca de informações baseado na vizinhança local também é chamado de  e apresenta vantagens em relação à topologia global [21]. Devido à distância lógica existente entre as partículas, a topologia local mostra-se bastante eficiente em não convergir

rapidamente para mínimos locais, pois uma partícula demorará a receber a informação sobre a

Com isso, diferentes regiões do espaço de busca podem ser exploradas ao mesmo tempo, sobre o qual as partículas procuram se mover em direção à melhor solução encontrada pelos seus vizinhos.

Embora a troca de informação entre as partículas seja mais lenta, esta estrutura provê uma melhor qualidade de soluções para problemas multimodais em comparação ao mesmo provido pela topologia global [21].

Na topologia global, conforme ilustrado na figura 6.b, o enxame está organizado em formato estrela e todas as partículas estão conectadas entre si. Esta topologia utiliza o mecanismo de vizinhança global, também denominado de g_{best} para a troca de informação.

Ao contrário da topologia anteriormente descrita, esta topologia permite uma convergência mais acelerada, visto que a informação da melhor posição é disseminada rapidamente entre todas as partículas do enxame. A comunicação é análoga a uma imensa comunidade onde todas as decisões tomadas são rapidamente conhecidas por todos os membros. Entretanto, o risco do algoritmo convergir a um mínimo local, não explorando devidamente todo o espaço de busca, é grande. Esta estrutura provê maior rapidez na busca da melhor solução para problemas multimodais, porém não garante a qualidade da solução obtida. Nestes casos o algoritmo pode atingir um mínimo local, devido a sua convergência precoce.

Diversos aspectos de diversas topologias foram analisados por J. Kennedy e R.Mendes originando novas propostas na modificação do comportamento do enxame [20]. O foco dessas modificações demonstra que a melhora de desempenho no algoritmo está relacionada diretamente em como as partículas se interagem dinamicamente dentro do enxame.

Diante dessas premissas, novas topologias foram propostas para o algoritmo *PSO*, inclusive com estruturas compostas da combinação da topologias global e local [20],[23]. Este trabalho propõe-se em apresentar uma nova topologia se inspirando em comunicação local com comutação dinâmica da vizinhança. A nova topologia é baseada na composição de vários anéis (*Multi-Ring*) e será explicada em detalhes no próximo capítulo.

Capítulo 4

Topologia Multi-Ring

Este capítulo visa apresentar a nova topologia *Multi-Ring* proposta para o algoritmo de otimização por enxame de partículas. São detalhados conceitos teóricos, propriedades específicas e a estrutura relacionada à nova topologia.

4.1 Estrutura do Multi-Ring

A nova topologia proposta neste trabalho para o *PSO* é denominada de *Multi-Ring*. Esta topologia é inspirada na topologia local e consiste na combinação de múltiplos anéis de partículas, onde cada partícula pode se comunicar com algumas partículas de anéis adjacentes.

A comunicação entre as partículas de cada anel é local, isto é, cada partícula do enxame nessa topologia comunica com seus vizinhos diretos. A diferença para a topologia local se baseia na composição de diversos anéis, fazendo com que a capacidade de exploração de todo enxame melhore, porém ao mesmo tempo forneça maior qualidade à solução final obtida pelo algoritmo.

Para a composição dos diferentes anéis da nova topologia, o número de partículas em cada anel deve ser o mesmo. Logo, cada partícula pertencente a um determinado anel é diretamente conectado a outras partículas vizinhas posicionadas no mesmo anel e anéis vizinhos. A Figura 7 apresenta visualmente a estrutura do *Multi-Ring*, onde cada partícula é representada por pontos.

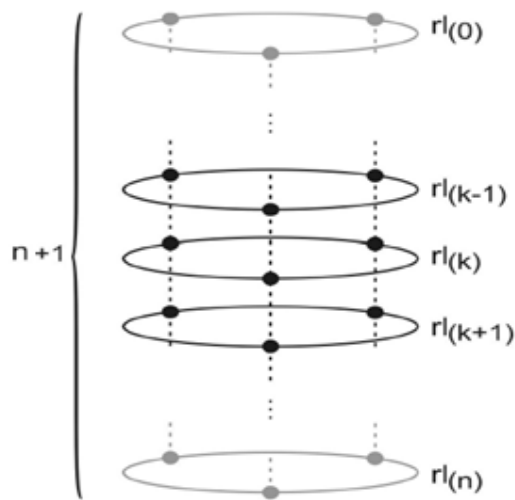


Figura 7. Estrutura da Topologia *Multi-Ring*.

Dado um conjunto de $n+1$ anéis rl , com k sendo o índice do anel, cada anel do *Multi-Ring* é representado por $rl_{(k)}$. Cada partícula do anel $rl_{(k)}$ é representado pelo índice i , ou seja, $rl_{(k)(i)}$. Considerando tais representações, cada partícula está diretamente conectada às partículas $rl_{(k-1)(i)}$ e $rl_{(k+1)(i)}$ do anel inferior e superior respectivamente além dos seus vizinhos diretos do próprio anel. Portanto, pode-se concluir que a estrutura do *Multi-Ring* é formada por anéis de partículas inter-conectados, onde durante a execução do algoritmo, cada partícula $rl_{(k)(i)}$ irá trocar informações localmente com os seus vizinhos diretos formado pelo conjunto $\{rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k-1)(i)}, rl_{(k+1)(i)}\}$.

Uma observação em especial deve ser considerada para o primeiro $rl_{(0)}$ e último anel $rl_{(n)}$ da topologia. A comunicação das partículas situadas nesses dois anéis é parcialmente limitada, visto que ambos não possuem a vizinhança de anéis completa, isto é, têm apenas um único anel diretamente conectado a eles. Portanto, a troca de informações para uma partícula pertencente ao anel $rl_{(0)}$ é limitada ao conjunto $\{rl_{(1)(i+1)}, rl_{(0)(i-1)}, rl_{(0)(i+1)}\}$. O mesmo vale ao anel $rl_{(n)}$ que é representado pelo conjunto $\{rl_{(n-1)(i-1)}, rl_{(n)(i-1)}, rl_{(n)(i+1)}\}$. Pode-se notar por estes conjuntos que as partículas têm uma comunicação semi-completa devido à inexistência de um dos anéis vizinhos.

4.2 Comunicação da topologia

Um dos grandes problemas enfrentados pelo *PSO* na busca da melhor solução é a estagnação. Em problemas multimodais não lineares, que são caracterizados por várias regiões com mínimos locais, o enxame tende a convergir rapidamente e com isso tende a crescer a probabilidade de o enxame ficar preso em um mínimo local [20].

Um mecanismo proposto para evitar esse problema é a habilidade de rotação do anel de partículas. Assim, quando uma rotação é efetuada, a posição das partículas no anel é alterada e a estagnação em um mínimo local é evitada, forçando o enxame a se mover de uma região de mínimo local para outra, aumentando a exploração do espaço de busca entre diferentes regiões de mínimos locais.

No *Multi-Ring*, as partículas de diferentes anéis se comunicam entre si localmente. A partícula que pertence a um anel $rl_{(k)}$ troca informações com outros anéis $rl_{(k-1)}$ e $rl_{(k+1)}$ com o objetivo de transmitir a sua experiência através de um mecanismo social por vizinhança local. Logo, as informações originadas de um anel são transmitidas para outros anéis.

Como cada anel do *Multi-Ring* tem a habilidade de rotacionar a si mesmo, a rotação é disparada se o anel não melhorar a sua melhor posição encontrada, definida como *ringbest*, durante um certo número de iterações. No processo de rotação cada partícula do anel é trocado de posição, baseado no índice de deslocamento d e no número de partículas em um anel nl onde o seu índice final é alterada para $i = (i+d) \bmod (nl)$. Logo, com a partícula rotacionada, a sua nova vizinhança será composta por outras partículas: $\{rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k+1)(i+d)}, rl_{(k-1)(i+d)}\}$. Por este conjunto descrito, pode-se observar que a principal diferença em relação à antiga vizinhança está na substituição da comunicação agora com a nova partícula superior e inferior dos anéis vizinhos.

A Figura 8 ilustra um exemplo do processo de rotação. Inicialmente, a partícula E se comunica com os vizinhos $\{D,F,B,H\}$. Após a rotação, a vizinhança da partícula E é alterada para $\{D,F,A,G\}$. Também se pode observar que não é apenas a partícula E que altera sua vizinhança, todas as partículas do mesmo anel sofrem a mesma modificação. A rotação sobre um anel estagnado tem o objetivo de melhorar a capacidade de convergência de todo o enxame.

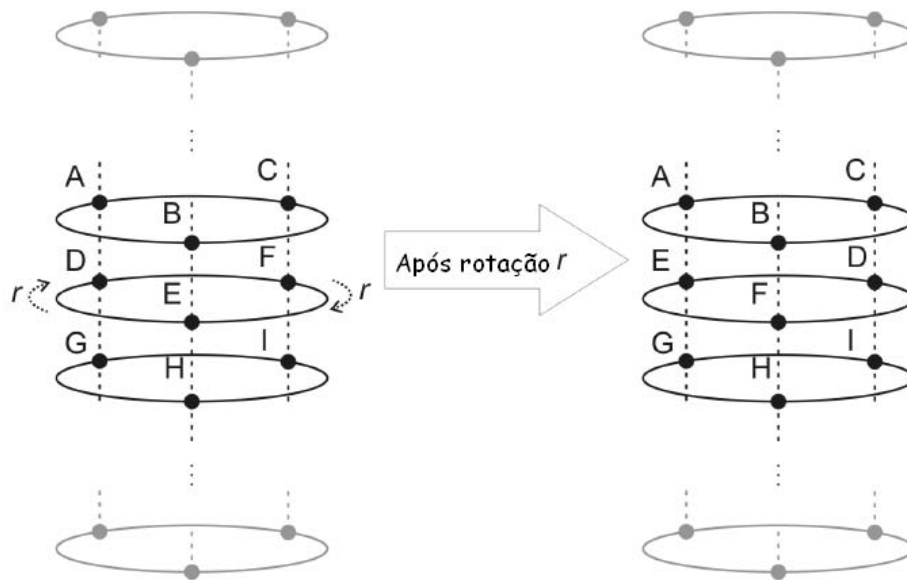


Figura 8. Exemplo de rotação da Topologia *Multi-Ring*.

Um mecanismo importante também pertinente ao processo de rotação de anéis é a escolha do gatilho de disparos Tr (*trigger*). Foi adotado como critério de disparo de rotação para a *Multi-Ring* o número de iterações em que a melhor solução encontrada por um anel específico não varia. Para exemplificar o processo descrito, considerando um anel $rl_{(k)}$, em que a melhor solução encontrada por este esteja estática, ou seja, com o mesmo valor inalterado durante Tr interações, o anel logo após a Tr iteração será rotacionado com um deslocamento de d partículas.

A estrutura proposta do *Multi-Ring* evidencia o uso da capacidade de exploração da comunicação local aliado ao uso de mecanismos de rotação que evita a estagnação da solução encontrada pelo enxame durante o processo de execução do algoritmo. Na próxima seção é descrito o comportamento do algoritmo *PSO* com as partículas estruturadas em anéis propostas pela topologia *Multi-Ring*.

4.3 Algoritmo PSO com a Topologia Multi-Ring

O algoritmo de otimização por enxame de partículas sofre algumas modificações com o esquema de comunicação proposto pelo *Multi-Ring*. Diferente das topologias global e local, em que as partículas são agrupadas em uma única estrutura, a topologia *Multi-Ring* agrupa as partículas em múltiplos anéis. Com isso, algumas etapas específicas devem ser agora consideradas no roteiro de execução do algoritmo.

No *PSO* básico global, uma destas etapas é a definição do *pbest* e *gbest* inicial, que representam a melhor posição encontrada pela partícula e pelo enxame, respectivamente. A maneira como é realizada a escolha e atualização do *pbest* não é influenciada com o uso da topologia *Multi-Ring*, visto que como se trata do conhecimento individual da partícula, todas as partículas do enxame definem o *pbest* de forma semelhante.

Ao contrário do *gbest*, no *Multi-Ring* é definido *ringbest* como a melhor posição encontrada por cada anel. Logo, a escolha do *ringbest* deve considerar apenas o anel de partículas, diferente das topologias em que as partículas são agrupadas sobre uma única estrutura como, por exemplo, a global e a local.

A definição da melhor solução se dá em duas etapas. Como o *Multi-Ring* está estruturado em múltiplos anéis, a primeira etapa consiste na escolha da melhor posição encontrada por cada anel da topologia (*ringbest*). Definidos os *ringbests* locais para cada anel, a segunda etapa incide na seleção do melhor *ringbest* referente a toda topologia. A escolha é sobre os *ringbests* definidos na primeira etapa, onde a melhor posição encontrada dentre todos os anéis será a melhor solução final para o algoritmo.

O processo de atualização do *ringbest* também é semelhante ao roteiro descrito acima, apenas com a diferença que em vez de ser definido o valor inicial para o *ringbest*, agora o *ringbest* final encontrado pelo anel é comparado com o valor encontrado na iteração anterior e atualizado se for considerada a melhor posição. A Figura 9 ilustra este processo da definição e atualização do *ringbest/pbest* durante a execução do algoritmo.

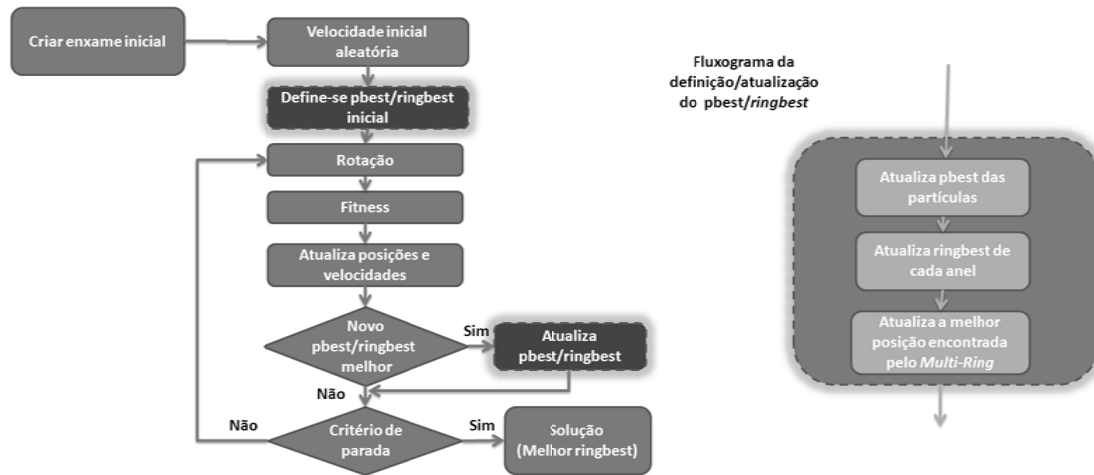


Figura 9. Algoritmo *PSO – Multi-Ring* com a definição do *pbest* e *ringbest*.

Outra etapa realizada no algoritmo com *Multi-Ring* e que não está incluso no algoritmo básico é o esquema de rotação. A Figura 10 apresenta em detalhes a inclusão desse novo esquema no roteiro do algoritmo *PSO*. Pode-se observar que essa etapa é realizada antes da avaliação do *fitness*. Primeiramente, é realizada a verificação do gatilho de rotação () e de acordo com o critério de disparo escolhido para o algoritmo, caso seja satisfeito, o disparo é realizado e o anel é rotacionado. Caso contrário, o critério de disparo do anel é incrementado e o fluxo de execução do algoritmo segue normalmente. Para o caso dos anéis serem rotacionados, o critério de disparo retorna ao seu estado inicial antes de seguir o fluxo restante de execução. Nesse esquema, a avaliação do gatilho de disparo é feita para cada anel do *Multi-Ring*. Portanto, fica claro que cada anel possui seu gatilho de rotação .

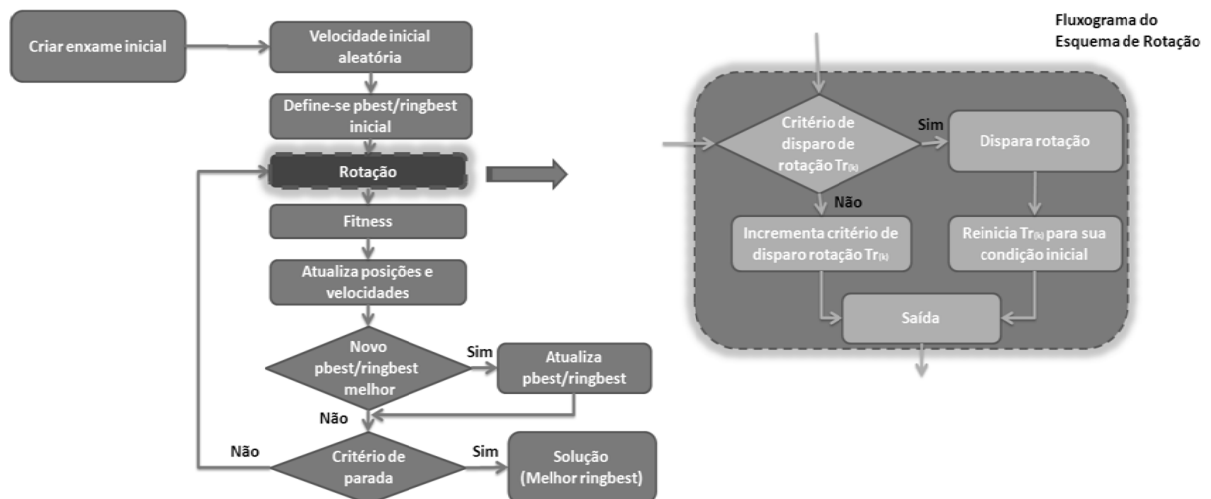


Figura 10. Algoritmo *PSO- Multi-Ring* com a etapa de rotação inclusa.

Capítulo 5

Experimentos

Este capítulo visa apresentar o arranjo experimental e a análise dos resultados obtidos utilizando o *PSO* com a topologia *Multi-Ring*. Também é realizada uma comparação com o *PSO* com topologias local e global. Serão descritos os detalhes dos arranjos experimentais, as funções de teste utilizadas para avaliação de desempenho, parâmetros de simulação e por fim a apresentação e discussão dos resultados obtidos.

5.1 Arranjo Experimental

Nesta seção são detalhados todos os requisitos necessários para a realização dos experimentos. Isso inclui informações sobre o simulador desenvolvido para o experimento, as funções de teste para avaliação de desempenho e os parâmetros de simulação utilizados.

5.1.1 Simulador *PSO Simulation Shell*

Para a realização dos experimentos, foi necessário o desenvolvimento de uma ferramenta de simulação de algoritmos *PSO*. O propósito do simulador é de facilitar a manipulação dos parâmetros de entrada das simulações e gerar resultados através de relatórios pré-definidos. Além de servir como um repositório para novas implementações e variações do algoritmo *PSO*.

Para este trabalho, a topologia *Multi-Ring* foi implementada e adicionada à interface do simulador para fins de simulação. A Figura 11 apresenta a interface do simulador do algoritmo *PSO – MultiRing* em execução para minimizar um problema de otimização.

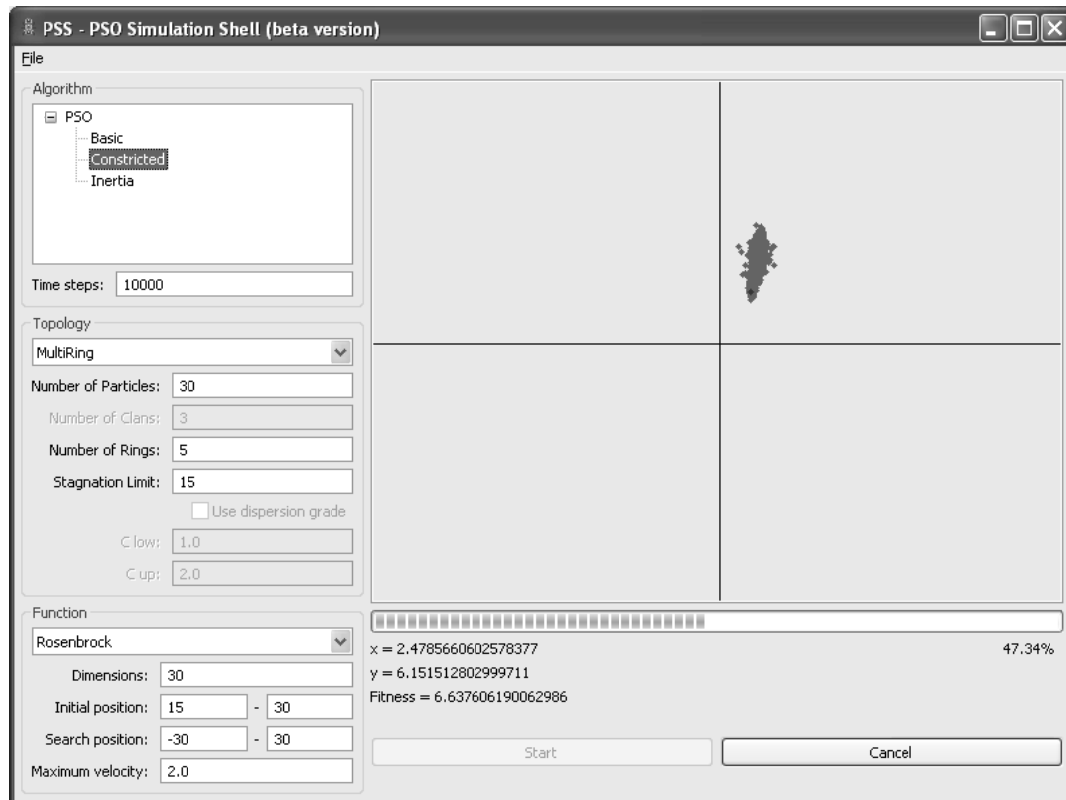


Figura 11. Interface do Simulador *PSO* com a topologia *Multi-Ring*.

5.1.2 Funções – Teste

Com o objetivo de avaliar o desempenho do *PSO* com a topologia *Multi-Ring*, foram selecionadas algumas funções matemáticas de otimização. Essas funções são bastante utilizadas para testes de avaliação (*benchmarks*) de algoritmos de otimização em encontrar a solução ótima [19]. Para este trabalho, foram escolhidas cinco funções envolvendo problemas de minimização.

Duas destas funções denominadas, *Rosenbrock* (8) e *Schwefel 1.2* (12) representam problemas de otimização não lineares mais simples, que são caracterizadas por terem a solução ótima localizada em uma região do espaço de busca bastante restrita. As outras três funções denominadas, *Rastrigin* (9), *Griewank* (10) e *Ackley* (11) são funções multi-modais não lineares caracterizadas pela sua alta complexidade e múltiplos mínimos locais. A Figura 12 abaixo ilustra matematicamente as funções de teste utilizadas para as simulações deste experimento.

$$F_{Rosenbrock}(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right], \quad (8)$$

$$F_{Rastrigin}(x) = 10n + \sum_{i=1}^n \left[x_i^2 - 10 \cos(2\pi x_i) \right], \quad (9)$$

$$F_{Griewank}(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad (10)$$

$$F_{Ackley}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, \quad (11)$$

$$F_{Schwefel\ 1.2}(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (12)$$

Figura 12. Funções de teste utilizadas nos experimentos.

Para cada uma dessas funções, foram padronizados as fronteiras do espaço de busca, a região de inicialização das posições das partículas, o número de dimensões e o a melhor solução do espaço de busca. Esses valores foram obtidos através de estudos e testes de avaliação de desempenho do algoritmo *PSO* e são bastante utilizados na literatura para configuração dos parâmetros das simulações de comparação de novas propostas do *PSO* com o algoritmo padrão. Nas simulações realizadas neste experimento, foram utilizados tais parâmetros recomendados, que são apresentados na Tabela 2 [25].

Tabela 2. Parâmetros da função-teste

<i>Rosenbrock</i> (8)	$-30 \leq x_i \leq 30$	$15 \leq x_i \leq 30$	30	$1,0^D$
<i>Rastrigin</i> (9)	$-5,12 \leq x_i \leq 5,12$	$2,56 \leq x_i \leq 5,12$	30	$0,0^D$
<i>Griewank</i> (10)	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$	30	$0,0^D$
<i>Ackley</i> (11)	$-32 \leq x_i \leq 32$	$16 \leq x_i \leq 32$	30	$0,0^D$
<i>Schwefel 1.2</i> (12)	$-100 \leq x_i \leq 100$	$50 \leq x_i \leq 100$	30	$0,0^D$

5.1.3 Parâmetros de simulação

Todas as simulações foram realizadas utilizando o algoritmo *PSO* com o fator de encolhimento (*constriction factor*) com os parâmetros de confiança $C_1 = C_2 = 2,05$. Conforme explicado anteriormente na seção 3.1.3, a escolha do algoritmo *PSO* modificado com o fator de enxugamento para as simulações foi baseada no grande desempenho obtido pelo mesmo, e que desde então, tem se tornado um padrão para avaliação e testes de desempenho de variações do algoritmo *PSO*.

Para a avaliação do *PSO* com a topologia *Multi-Ring*, os mesmos experimentos também foram realizados com o *PSO* com topologias global e local. Com isso, objetiva-se demonstrar a melhora de desempenho do *PSO* utilizando a nova topologia em comparação com as utilizadas na literatura padrão.

Todo o exame é inicializado aleatoriamente na sub-região do espaço de busca disponível longe da solução ótima em todas as dimensões de acordo com a função a ser minimizada. Tais parâmetros de inicialização podem ser vistos na Tabela 2 descrito na seção 5.1.2. Para todas as simulações foram utilizadas 30 partículas.

Para a topologia *Multi-Ring*, foi considerada a variação do número de anéis (n). Logo, considerando n anéis, o número de partículas por anel (nl) será $30/n$. Por exemplo, caso uma simulação tenha como parâmetro 5 anéis, o número de partículas por anel será 6, ou seja, 5 anéis de 6 partículas. Para o esquema de rotação, foi considerado como gatilho de disparo $Tr_{(k)}$ o número de iterações em que o anel encontra-se estagnado, conforme explicado na seção 4.2. O deslocamento d de rotação foi definido com o valor igual à metade do número de partículas por anel ($d = nl / 2$).

Todas as simulações foram executadas com 10.000 iterações. Para cada simulação foi utilizado o algoritmo. Após 30 simulações, para cada configuração de parâmetros, foram calculados a média e desvio padrão do *fitness* obtidos durante a execução de tais simulações.

Na próxima seção, serão detalhados os resultados obtidos pelos experimentos realizados com a topologia *Multi-Ring*, além das topologias global e local.

5.2 Resultados

Esta seção apresenta os resultados obtidos pelos experimentos, utilizando as funções de teste descritas anteriormente na seção 5.1.2. Esta seção está dividida em duas partes. Na primeira parte serão apresentados os resultados das simulações baseadas na variação do número de anéis da topologia *Multi-Ring*. A segunda parte apresenta os resultados das simulações variando o número de iterações do gatilho de disparo de rotação Tr usando as melhores configurações obtidas na primeira parte dos experimentos.

5.2.1 Resultados pela variação do número de anéis

Cinco diferentes configurações de *Multi-Ring* foram utilizadas nas simulações para fins de comparação. O número de anéis presentes na topologia foi variado para que fosse possível analisar o impacto deste parâmetro. Foram testados os seguintes valores: 2, 3, 5, 6 e 10 anéis. A escolha desses valores foi baseado em um critério arbitrário onde se considerou razoáveis tais números de anéis com a premissa de investigar a influência do número de anéis no desempenho do algoritmo *PSO*. Todas as simulações dessa primeira parte de experimentos foram realizadas com o valor do gatilho de disparo de rotação ($Tr_{(k)}$) igual a 20.

Basicamente, todos os resultados estão distribuídos em tabelas compostas por cinco colunas. Uma coluna para descrição da configuração usada, duas colunas com *fitness* médio e outras duas com o desvio padrão médio. Estes pares de colunas apresentam os valores obtidos para 5000 e 10000 iterações respectivamente. Também são apresentados gráficos para cada configuração, ilustrando a taxa de convergência em função do número de iterações até a solução ótima.

A Tabela 3 apresenta os resultados obtidos para a função de teste *Rosenbrock* utilizando as configurações baseadas no número de anéis do *Multi-Ring* e as topologias global e local. Pode-se observar que todas as simulações com as configurações *Multi-Ring* obtiveram melhores resultados do que os obtidos pelas topologias global e local. Para as simulações com 5000 iterações, o melhor *fitness* encontrado foi com a configuração *Multi-Ring* com 2 anéis. Com 10000 iterações, a configuração *Multi-Ring* com 5 anéis obteve o melhor resultado.

A figura 13 apresenta a curva de convergência em função do número iterações para as mesmas configurações. Pode-se observar que para todas as configurações, a taxa de convergência não teve grandes diferenças, apenas salientando que em comparação à topologia local, todas as

configurações *Multi-Ring* obtiveram uma convergência mais rápida. Os resultados também destaque a configuração de 5 anéis com 1000 interações com fitness médio bem melhor em comparação aos obtidos pela topologia local e global.

Tabela 3. Resultados da função *Rosenbrock* após 5000 e 10000 iterações.

Configuração	Fitness			
	5000 iterações		10000 iterações	
	Média	Desvio Padrão	Média	Desvio Padrão
2 anéis	20,0972	25,374	4,8107	4,248
3 anéis	30,6683	32,253	4,7912	4,406
5 anéis	26,4892	34,302	4,7407	4,311
6 anéis	24,4892	32,124	5,5545	6,181
10 anéis	21,0950	28,729	5,2740	5,586
Local	31,0843	30,853	14,2054	22,446
Global	23,3124	29,058	8,2631	16,680

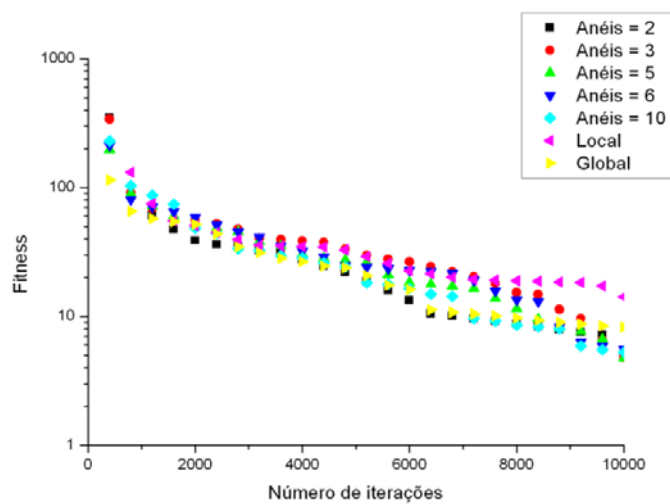


Figura 13. Resultados da função *Rosenbrock* exibindo as curvas de convergência para várias configurações.

A Tabela 4 apresenta os resultados obtidos para a função de teste *Rastringin* utilizando as soluções obtidas pelas simulações com as configurações do *Multi-Ring* tiveram melhores resultados em comparação com as mesmas obtidas pelas topologias local e global. A configuração *Multi-Ring* que obteve melhor desempenho foi a de 2 anéis, com *fitness* médio de 31,5401 e 28,8540 em 5000 e 10000 iterações, respectivamente.

A Figura 14 apresenta a curva de convergência em função do número iterações para as mesmas configurações. Pode-se observar que todas as configurações baseadas no *Multi-Ring* obtiveram a taxa de convergência durante a execução do *PSO* com melhor desempenho em comparação aos resultados obtidos pelas topologias global e local. Isso ratifica os resultados da Tabela 4, que demonstra a configuração *Multi-Ring* com 2 anéis como a configuração que obteve a melhor convergência do *fitness*.

Tabela 4. Resultados da função *Rastringin* após 5000 e 10000 iterações.

Configuração	Fitness			
	5000 iterações		10000 iterações	
	Média	Desvio Padrão	Média	Desvio Padrão
2 anéis	31,5401	8,451	28,8540	7,786
3 anéis	31,9058	7,208	38,6707	10,689
5 anéis	32,9058	7,208	31,904	7,2097
6 anéis	32,0377	7,3084	31,7723	6,9863
10 anéis	35,9511	6,409	35,6533	7,154
Local	42,6643	8,425	42,5422	8,383
Global	53,7608	15,394	53,7608	15,394

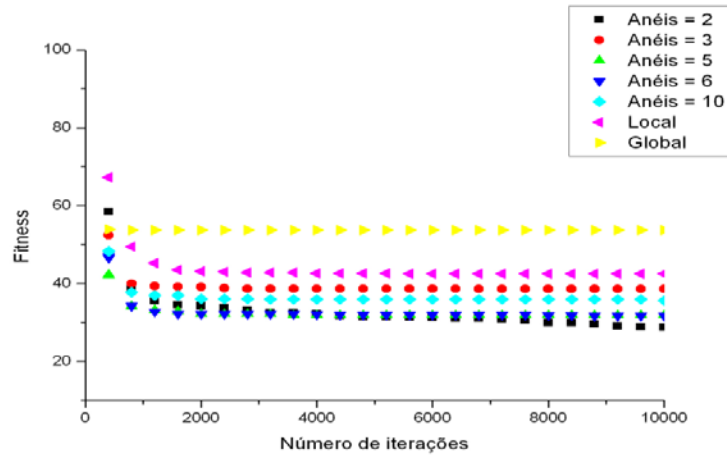


Figura 14. Resultados da função *Rastrigin* exibindo as curvas de convergência para várias configurações.

A Tabela 5 apresenta os resultados obtidos para a função de teste *Griewank* utilizando as configurações baseadas no número de anéis do *Multi-Ring* e nas topologias global e local. A função *Griewank* é uma função multi-modal de alta complexidade, onde a topologia local obtém os melhores resultados, convergindo mais rápido e assim, evitando mínimos locais. A simulação com a topologia local obteve melhor convergência do *fitness*, com o valor final após de 10000 iterações de 0,002381. O melhor resultado obtido pelo *Multi-Ring* foi com a configuração de 2 anéis, obtendo um *fitness* médio de 0,004927. Pela função *Griewank* ser uma função com diversas regiões de mínimos locais, topologias em que a troca de informação é mais lenta para se propagar têm um grande desempenho nesse tipo de problema. Logo, o *Multi-Ring* se propõe ser uma topologia equilibrada em que apresenta ainda boas soluções independente da complexidade ou dimensionalidade do problema a ser tratado.

A Figura 15 apresenta os traços de convergência até a melhor solução em função do número de iterações. Pode-se observar que para esta função, as configurações *Multi-Ring* não apresentaram uma rápida convergência em comparação aos apresentados pelas topologias global e local. Entretanto, apesar da lenta convergência as configurações *Multi-Ring* obtiveram resultados satisfatórios equivalentes aos valores encontrados pelas topologias global e local, durante a execução das últimas 2000 iterações.

Tabela 5. Resultados da função *Griewank* após 5000 e 10000 iterações.

Configuração	5000 iterações		10000 iterações	
	Média	Desvio Padrão	Média	Desvio Padrão
2 anéis	0,00492	0,00636	0,00492	0,0063
3 anéis	0,00730	0,00735	0,00640	0,0064
5 anéis	0,00771	0,01122	0,00771	0,0112
6 anéis	0,00730	0,00813	0,00730	0,0081
10 anéis	0,00501	0,00664	0,00484	0,0064
Local	0,00238	0,00494	0,002381	0,0049
Global	0,02643	0,03424	0,02643	0,0342

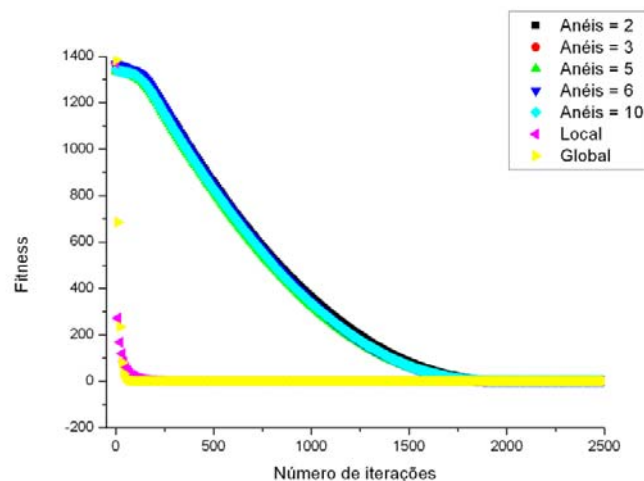


Figura 15. Resultados da função *Griewank* exibindo as curvas de convergência para várias configurações.

A Tabela 6 apresenta os resultados obtidos para a função de teste *Ackley* utilizando as configurações baseadas no número de anéis do *Multi-Ring* e nas topologias global e local. Todas as configurações obtiveram resultados bem parecidos sem muitas variações. Isso ocorre, pelo fato do algoritmo *PSO* conseguir convergir rapidamente com poucas iterações até uma solução ótima.

Através de uma análise mais detalhada dos resultados obtidos demonstrado pelo gráfico

apresentaram nas primeiras iterações uma convergência mais lenta em comparação aos resultados obtidos com as topologias global e local. A possível causa dessa convergência é a comunicação entre anéis, que faz com que as informações se propaguem gradativamente até todos os anéis, provocando assim uma convergência até a solução ótima de forma assintota. Entretanto, após 200 iterações as configurações *Multi-Ring* apresentaram melhores resultados, convergindo até a solução ótima com melhores valores de *fitness*. Isso demonstra que a topologia *Multi-Ring* caracterizada por não ter uma convergência tão rápida e nem tão lenta, consegue obter melhores resultados. Por fim, pode-se observar que o *Multi-Ring* com 2 anéis, foi a configuração que obteve os melhores valores de *fitness*.

Tabela 6. Resultados da função *Ackley* após 5000 e 10000 iterações.

Configuração	Fitness			
	5000 iterações		10000 iterações	
	Média	Desvio Padrão	Média	Desvio Padrão
2 anéis	19,8124	0,0299	19,8118	0,0293
3 anéis	19,8226	0,0201	19,8218	0,0196
5 anéis	19,8186	0,0240	19,8186	0,0240
6 anéis	19,8147	0,0309	19,8142	0,0305
10 anéis	19,8177	0,0189	19,8177	0,0189
Local	19,9364	0,0180	19,9154	0,0235
Global	19,9230	0,0218	19,9141	0,0249

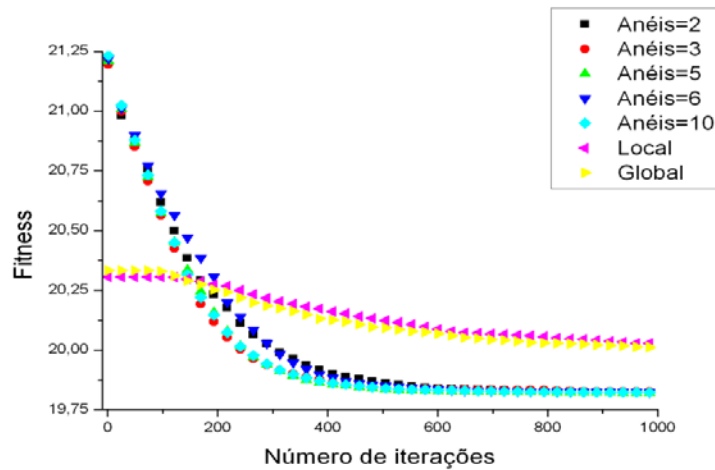


Figura 16. Resultados da função *Ackley* exibindo as curvas de convergência para várias configurações.

A Tabela 7 apresenta os resultados obtidos para a função-teste *Schwefel 1.2* utilizando as configurações baseadas no número de anéis do *Multi-Ring* e nas topologias global e local. Pode-se observar que a topologia global obteve os melhores resultados em comparação às outras configurações, pois a função *Schwefel 1.2* é uma função complexa em que algoritmos de exploração em profundidade obtêm melhor desempenho. Já para o *Multi-Ring*, todas as suas configurações obtiveram melhores valores de *fitness* em relação aos mesmos obtidos pela topologia local, provando ter um melhor desempenho, já que seu mecanismo de rotação evita a estagnação prematura das partículas em mínimos locais.

A Figura 17 apresenta a taxa de convergência em função do número iterações para as mesmas configurações. Pode-se observar que a convergência é mais lenta para as configurações *Multi-Ring* em comparação às topologias global e local. A possível causa seria devido à comunicação entre anéis, que faz com que as informações se propaguem gradativamente até todos os anéis, provocando assim uma convergência até a solução ótima de forma assintota. Entretanto, as configurações *Multi-Ring* chegam a valores de *fitness* equivalentes às outras topologias após a execução das primeiras 300 iterações do algoritmo.

Configuração	Fitness			
	5000 iterações		10000 iterações	
	Média	Desvio Padrão	Média	Desvio Padrão
2 anéis	1,29E-02	0,0090	3,86E-7	5,6E-07
3 anéis	2,92E-03	0,0024	1,69E-8	1,4E-08
5 anéis	1,74E-03	0,0015	7,99E-9	7,6E-09
6 anéis	2,60E-03	0,0032	1,18E-8	1,4E-08
10 anéis	3,67E-03	0,0033	3,58E-8	4,1E-08
Local	9,23E-01	0,6195	2,88E-4	0,0033
Global	8,17E-09	1,5E-8	1,3E-20	3,1E-20

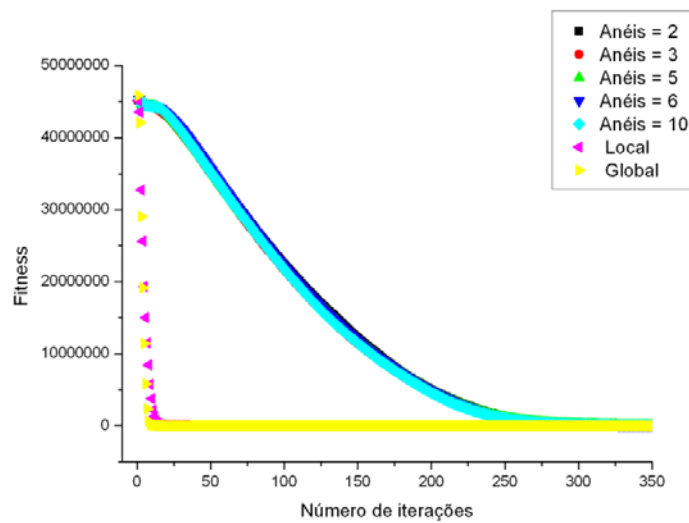


Figura 17. Resultados da função *Schwefel 1.2* exibindo as curvas de convergência para as várias configurações.

5.2.2 Resultados pela variação do gatilho de disparos de rotação

Nesta seção são apresentados os resultados dos experimentos com a configuração *Multi-Ring* variando o gatilho de disparos de rotação. A configuração *Multi-Ring* utilizada nessa fase de experimentos é escolhida baseada na configuração que obteve resultados satisfatórios em maior parte dos experimentos realizados e apresentados na seção 5.2.1. Além dos resultados obtidos, outro critério importante para escolha da configuração é a estrutura da topologia, com o objetivo de ver a influência da variação do gatilho de rotação no desempenho da execução do algoritmo.

A configuração *Multi-Ring* escolhida para esse experimento, de acordo com os resultados obtidos, é o *Multi-Ring* com 5 anéis. Esta configuração também tem a estrutura mais distribuída dentre as simuladas. Com base nessas informações, serão apresentados os resultados obtidos utilizando a configuração *Multi-Ring* com 5 anéis variando o número de iterações, que define o critério do gatilho de disparos de rotação Tr (*trigger*), conforme explicado na seção 4.2.

Foram selecionados como gatilhos de rotação os valores: 5, 10, 20, 50 e 100, representando o número de iterações em que um anel do *Multi-Ring* espera sem rotacionar sem que uma melhora ocorra. Estes valores foram selecionados aleatoriamente com o objetivo de investigar a influência do gatilho de disparos no desempenho do *PSO*. Foram utilizadas as funções de teste utilizadas na seção 5.2.1, focando na minimização das mesmas. Todas as simulações foram executadas com 10000 iterações com os *fitness* médios apresentados para fins de comparação.

A Tabela 8 apresenta os resultados obtidos pelas simulações com as configurações *Multi-Ring* para as funções de teste. O valor numérico em cada coluna é representado pelo *fitness* médio e o desvio padrão, este representado entre parênteses.

<i>Tr</i>	Fitness (<i>Multi-Ring</i> 5 anéis)				
	<i>Rosenbrock</i>	<i>Rastrigin</i>	<i>Griewank</i>	<i>Ackley</i>	<i>Schwefel 1.2</i>
5	3,84	34,49	0,0089	19,82	1,3E-08
	(3,04)	(11,3)	(0,012)	(0,02)	(1,6E-08)
10	4,14	35,81	0,0059	19,81	1,8E-8
	(3,83)	(6,64)	(0,008)	(0,02)	(1,6E-8)
20	4,740	31,90	0,0077	19,81	7,99E-9
	(4,31)	(7,20)	(0,011)	(0,02)	(7E-09)
50	6,14	34,02	0,0054	19,81	2,8E-8
	(7,54)	(7,66)	(0,007)	(0,01)	(3,0E-8)
100	3,96	39,03	0,0077	19,81	4,3E-8
	(4,55)	(8,81)	(0,012)	(0,02)	(6,0E-8)

Pelos resultados apresentados na Tabela 8, pode-se concluir que a configuração *Multi-Ring* de 5 anéis com o valor do gatilho de disparos de rotação igual a 20 ($Tr = 20$) obteve os resultados mais expressivos e regulares para todas as funções testadas. Outra observação que pode ser extraída pelos resultados demonstrados na Tabela 8 é o fato de que grandes valores para o gatilho de rotação reduzem o poder de exploração da topologia *Multi-Ring* em algumas funções. Isso acontece devido ao fato de que a topologia fica em preso em mínimos locais por diversas iterações até que seja rotacionada. Essa rotação tardia provoca a falta de exploração e conseqüentemente um pior desempenho na execução do algoritmo *PSO*.

De forma semelhante ocorre para valores muito baixos para o gatilho de rotação na maioria das funções testadas, conforme os resultados apresentados na Tabela 8. O baixo desempenho se dá pelo fato de que como a rotação dos anéis do *Multi-Ring* ocorre de forma prematura, isto é, cedo demais. Isto faz com que as partículas se dispersem, provocando mudanças bruscas na trajetória de convergência das partículas.

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho se propõe em apresentar uma nova topologia para o algoritmo de otimização por enxame de partículas (*PSO*) com o objetivo de mostrar melhorias no desempenho em relação às topologias existentes, assim contribuindo com avanços nas pesquisas de técnica de otimização inteligentes.

Neste capítulo, serão feitas algumas considerações sobre os resultados dos experimentos realizados em relação à nova topologia proposta neste trabalho, como também sobre as contribuições aos estudos realizados na busca de novas variações do *PSO* e por fim os possíveis trabalhos futuros para extensão deste trabalho.

6.1 Contribuições

Fica claro que com o aparecimento de problemas cada vez mais complexos, é necessário fazer modificações nas técnicas de otimização a fim de se obter um melhor desempenho na busca de soluções adequadas. Este trabalho contribui com os estudos que vêm sendo realizados recentemente na busca por novas modificações e variações na heurística de otimização por enxame de partículas (*PSO*).

Dentre os diversos componentes modificados e adicionados no algoritmo original do *PSO*, a organização é uma maneira de torná-lo mais eficiente. Com isso, esse trabalho estende o algoritmo *PSO* com uma nova proposta de topologia, que se torna uma alternativa eficiente apresentando um melhor desempenho em comparação às topologias padrões adotadas na literatura, assim contribuindo com avanços nas pesquisas de técnicas de otimização. Outra

contribuição importante é a inclusão de novos mecanismos ao processo de execução do algoritmo

Tais contribuições são provadas pelos experimentos realizados que demonstram os estudos comparativos entre a nova topologia *Multi-Ring* e as topologias existentes por meio de simulações envolvendo a minimização de funções matemáticas, bastante utilizadas para avaliação de desempenho na literatura.

A topologia *Muli-Ring* se mostra ser bastante eficiente na resolução de problemas de alta dimensionalidade e complexidade. Com novos mecanismos introduzidos neste trabalho como o esquema de rotação e a troca de informações entre anéis adjacentes, novas variáveis podem ser consideradas para a configuração dos parâmetros do algoritmo *PSO* no processo de otimização e busca.

Outra principal contribuição é a introdução de novas etapas no algoritmo *PSO*. Com a inclusão do esquema de rotação e a introdução do ringbest, que informa a melhor posição encontrada pelo anel, o algoritmo *PSO* passa a ter novos fluxos no seu processo de execução. Esses fluxos podem ser facilmente adicionados ou replicados para outras variações do *PSO* sem muito esforço, já que este trabalho focou na modularização e extensão do algoritmo.

A realização deste trabalho também gerou a publicação de um artigo, com o título *Multi-Ring Particle Swarm Optimization*, submetido no Simpósio Brasileiro de Inteligência Artificial (SBIA) deste ano [26].

6.2 Conclusão

Este trabalho apresenta uma nova topologia denominada *Multi-Ring* inspirada em comunicação local com comutação dinâmica da vizinhança para o algoritmo *PSO*. Pelos experimentos realizados, conforme os resultados apresentados no capítulo 5, demonstra-se que o *PSO* modificado utilizando a topologia proposta obtêm melhores soluções do que os apresentados pelo *PSO* com a topologia local. Os resultados também provam que a estrutura aqui proposta é uma alternativa eficiente para busca de soluções de boa qualidade em problemas de otimização multi-modais que sofrem com estagnação.

As funções de teste utilizadas nos experimentos são caracterizadas por alta complexidade associados a não linearidades e múltiplos mínimos locais. Pelas funções de teste avaliadas, a

topologia *Multi-Ring* obteve melhores resultados em três das cinco funções, o que demonstra que

Pode-se concluir pelos resultados obtidos que a estrutura proposta pela topologia *Multi-Ring* apresenta uma melhoria na capacidade de exploração de todo o enxame e ao mesmo tempo fornecendo uma maior qualidade à solução final obtida pelo *PSO*. Logo, o algoritmo *PSO* com a topologia *Multi-Ring* consegue obter soluções de boa qualidade independente do problema a ser solucionado.

Com a adição do mecanismo de rotação, evita que o enxame fique preso em mínimos locais, forçando-os a se mover para novas regiões do espaço de busca a fim de fazer o algoritmo *PSO* convergir até a solução ótima. Logo, esse mecanismo se torna uma nova alternativa que pode ser utilizada em outras variações do *PSO*, inclusive em outras topologias, com objetivo de prevenir a estagnação do algoritmo em mínimos locais, sendo este um dos principais problemas que prejudicam o desempenho do *PSO* na busca da solução ótima.

Outro aspecto importante da topologia *Multi-Ring* é o gatilho de disparos de rotação. Para esses experimentos, que foi utilizado o número de iterações em que o anel de partículas ficou estagnado, os resultados apresentados demonstram que o gatilho de rotação influencia no desempenho do algoritmo *PSO*. A depender da complexidade do problema a ser solucionado, o gatilho de disparos é outra variável que pode ser modificada e pode auxiliar no trajeto de convergência do enxame pelo espaço de busca até a solução ótima.

Por fim, pode-se concluir que novos mecanismos e variáveis foram propostos com a introdução da topologia *Multi-Ring*. Os estudos iniciais validados pelos experimentos realizados neste trabalho demonstram que a topologia *Multi-Ring* é uma alternativa eficiente para resolução de problemas de busca e otimização em algoritmos de otimização por enxame de partículas. Sua topologia distribuída em múltiplos anéis com comutação dinâmica de vizinhança permite a difusão da informação entre as partículas de forma controlada por mecanismos aqui propostos a fim de evitar a estagnação e conseqüentemente a obtenção de soluções sub-ótimas, não adequadas para o problema tratado.

6.3 Trabalhos futuros

Como trabalhos futuros, pode-se citar a extensão da comunicação proposta pelo *Multi-Ring*, adicionando novos mecanismos de alteração de vizinhança. Isso inclui a possibilidade de extensão de troca de informações entre os vizinhos, permitindo que as partículas possam se comunicar não somente com seus vizinhos diretos, como também com seus n vizinhos locais. Isso vale também para a comunicação entre anéis, estendendo a troca de informações além dos anéis adjacentes para até n anéis da topologia.

Outro possível trabalho futuro é um aprofundamento nos estudos do gatilho de disparos de rotação, conforme visto na seção 4.2 Neste trabalho foi proposto como fator de deslocamento a metade do número de partículas por anel. Variações podem ser realizadas a fim de verificar a influência do gatilho de disparos no desempenho do algoritmo *PSO* na busca de soluções.

Outras modificações estruturais podem ser incorporadas ao *PSO-Multi-Ring* como variações na equação de atualização de velocidade das partículas a fim de incrementar a melhora de desempenho do *PSO*.

Outro possível ponto de melhoria é a realização de estudos e variações no *Multi-Ring* para a otimização da rapidez da taxa de convergência até a melhor solução, especialmente nas funções de teste apresentadas neste trabalho.

Um trabalho futuro também considerado é a modificação do esquema de rotação do *Multi-Ring* com disparos automáticos, baseados em desempenho da execução do algoritmo. Logo, vale verificar quais outros critérios de disparo de rotação além do número de iterações em que o anel ficou estagnado podem ser utilizados. Uma análise pode ser realizada em relação ao sentido da rotação. Neste trabalho, foi utilizado a rotação no sentido horário, porém podem ser realizados experimentos com variações no sentido da rotação.

Outro aspecto a ser considerado como trabalho futuro é a troca de informações e a determinação da vizinhança das partículas. Outros mecanismos podem ser utilizados para se determinar a nova vizinhança de uma partícula. Um desses mecanismos é a utilização de gaussianas para determinar a proximidade da partícula à solução ótima. Com isso as partículas tendem se aproximar mais das partículas com boas informações e se afastar das partículas com informações de baixa qualidade. Esse mecanismo pode ser modelado para construção de novos mecanismos de vizinhança na topologia *Multi-Ring*.



Por fim, outro trabalho futuro é a realização de estudos na solução de outros problemas de problemas reais, em que a solução ótima não é conhecida previamente.

Bibliografia

- [1] E. ZITZLER, K. DEB, AND L. THIELE. **Comparison of multiobjective evolutionary algorithms: Empirical results**. Evolutionary Computation, 8(2):173-195, 2000.
- [2] J. KNOWLES AND D. CORNE. **Approximating the non-dominated front using the pareto archived evolution strategy**. Evolutionary Computation, 8(2):149-172, 2000.
- [3] M. LAUMANN, E. ZITZLER, AND L. THIELE. **A unified model for multi-objective evolutionary algorithms with elitism**. Congress on Evolutionary Computation (CEC), páginas 46-53, Piscataway, NJ, 2000. IEEE Service Center.
- [4] J. KENNEDY AND R. EBERHART. **Swarm Intelligence**. Morgan Kaufmann Publishers, California, USA, 2001.
- [5] J. KENNEDY AND R. MENDES. **Population structure and particle swarm performance**. In Congress on Evolutionary Computation, Volume 2, páginas 1671-1676, Piscataway, NJ, Maio de 2002. IEEE Service Center.
- [6] J. KENNEDY AND R. EBERHART. **Particle swarm optimization**. In Proc. of the IEEE Int. Conf. on Neural Networks, páginas 1942-1948, Piscataway, NJ., 1995. IEEE Service Center.
- [7] D. M. GREIG. **Optimization**, capítulos 3-4. Longman Inc., New York, USA, 1980.
- [8] J. E. DENNIS, JR AND R. B. SCHNABEL. **Numerical Methods for Unconstrained Optimization and Nonlinear Equations**, capítulo 6, pp. 111-152. Prentice-Hall, 1983.
- [9] D. E. COLDBERG, **Genetic algorithms in search, optimization and machine learning**, Addison Wesley, 1989.
- [10] REEVES, C. R. **Modern Heuristic Techniques for Combinatorial Problems**. New York McGraw- Hill International. 1995. 320p.

- [11] SUCUPIRA, IGOR RIBEIRO. **Métodos heurísticos genéricos: metaheurísticas e** da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo.
- [12] BATISTA, MARÍA BELÉN MELIÁN. **Metaheuristic Optimization for WDM Network Network Planning.** 259f. Dissertação (Mestrado em Matemática) - Departamento de Estatística. 2004, Investigación Operativa y Computación, Universidad de Laguna.
- [13] POLTOSI, MAIRA. Programando Soluções para Problemas de Otimização Combinatória. No **Congresso Simulado de Técnicas de Programação**, São Leopoldo, 2006.
- [14] MILLONAS, M. M. **Swarms, phase transitions, and collective intelligence.** No: C. G. Langton, Ed., Artificial Life III. Addison Wesley, Reading, MA.1994.
- [15] POMEROY, P. **An Introduction to Particle Swarm Optimization.** 2003. Disponível em: <http://www.adaptiveview.com/articles/ipsoprnt.html>. Acesso em: 20 de abril de 2008.
- [16] MIRANDA, VLADIMIRO. **Computação Evolucionária Fenotípica.** Porto, 2005.
- [17] SHI, YUHUI. **Particle Swarm Optimization.** Em: Proceedings of the IEEE Neural Network Society. p. 8 – 13, 2004.
- [18] D. CORNE, M. DORIGO, AND F. GLOVER. **New Ideas in Optimization**, capítulo 25, pp. 379-387. McGraw Hill, 1999.
- [19] RUSS C. EBERHART AND Y. SHI. **Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization.** Em *Proceedings of the Congress on Evolutionary Computing*, San Diego, USA, pp. 84-89, 2000.
- [20] Y. SHI AND RUSS C. EBERHART. **A Modified Particle Swarm Optimizer.** Em *IEEE International Conference of Evolutionary Computation*, Anchorage, Alaska, Maio 1998.
- [21] P. ENGELBRECHT, **Fundamentals of Computational Swarm Intelligence**, John Wiley & Sons, 2005.
- [22] A. M. SUTTON, D. WHITLEY, M. LUNACEK, AND A. HOWE. **PSO and multi-funnel landscapes: how cooperation might limit exploration.** Em GECCO 2006 -: Proceedings of the 8th annual conference on Genetic and evolutionary computation, vol 1. Seattle, Washington, USA: ACM Press, 8-12 Julho. 2006, pp. 75–82.

- [23] P. N. SUGANTHAN. **Particle swarm optimizer with neighborhood operator**, em Washington D.C., USA: IEEE Press, 6-9 Jul. 1999, pp 1958-1962.
- [24] KIRKPATRICK, S. , GELATT JR., C. D., VECCHI, M. P. 1983, **Optimization by Simulated Annealing Science**, 220, N. 4598, pp 671-680.
- [25] D. BRATTON AND J. KENNEDY. **Defining a standard for particle swarm optimization**, em Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, Honolulu HI, Apr. 2007, pp 120-127.
- [26] Simpósio Brasileiro de Inteligência Artificial, (SBIA 2008). Website: [<http://www.sbia2008.ufba.br/>]



Apêndice A