



INTELIGÊNCIA COMPUTACIONAL

CAT SWARM OPTIMIZATION

Carolina Ferreira nº 2018018459

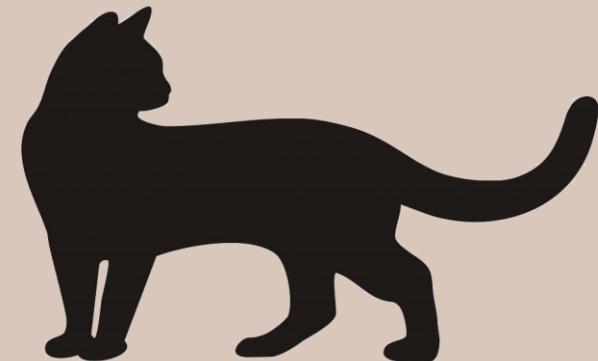
Eduardo Pina nº2017010181

1 INTRODUÇÃO

- A ciência computacional tem sido, nas últimas décadas, um tema de pesquisa frequente por parte de muitos cientistas dos mais diversos ramos científicos.
- A razão pela qual esta ciência é alvo de grande interesse está nos grandes avanços que tem proporcionado e ao conjunto de modelos matemáticos e técnicas de soluções numéricas desenvolvidas, que, utilizando computadores, permitem encontrar soluções ou até mesmo a solução ótima para problemas complexos.

Entre os mais variados algoritmos já desenvolvidos, destacam-se três:

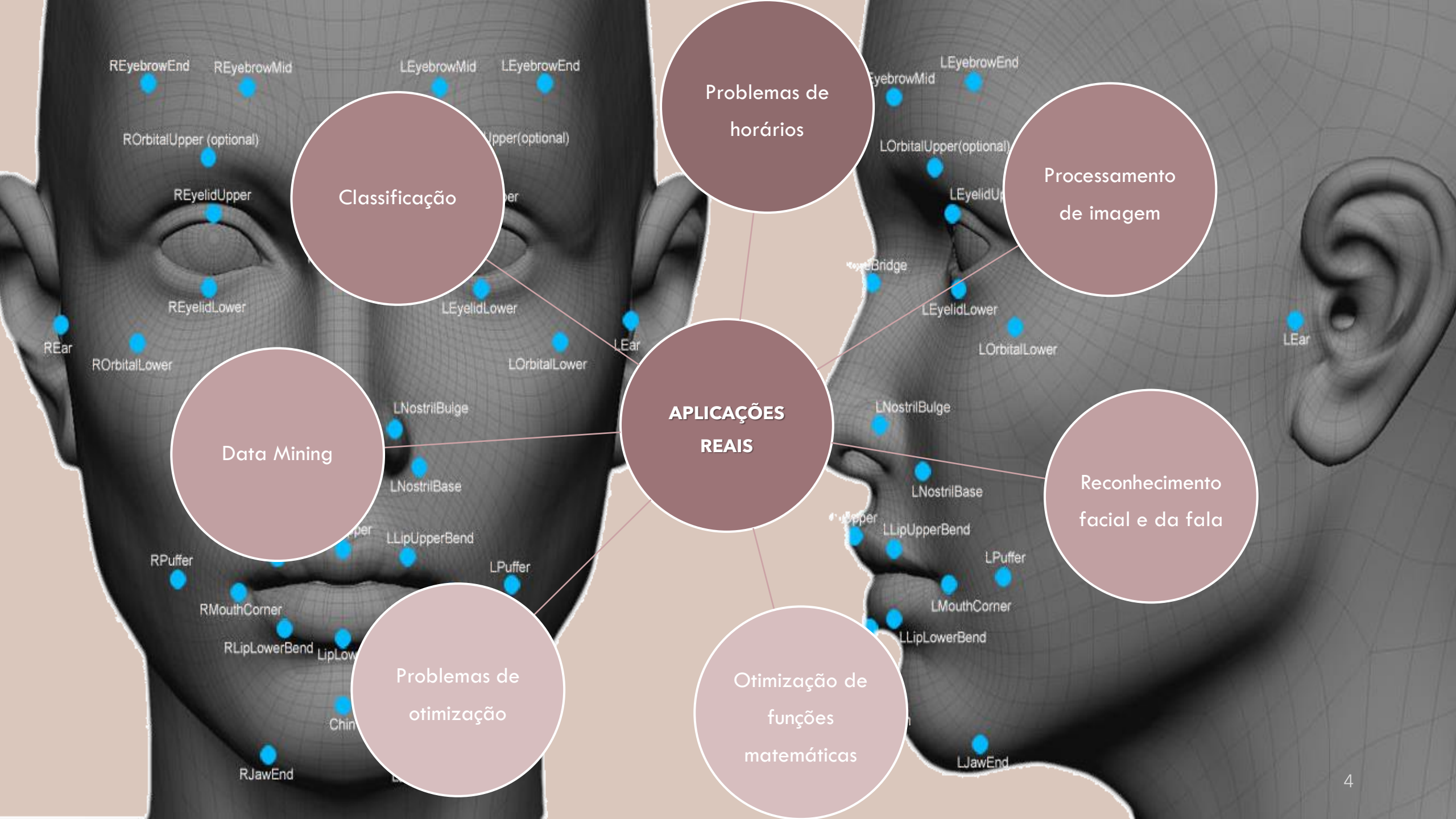
- Genetic Algorithm (GA)
- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)



1 INTRODUÇÃO

- No contexto do ACO e do PSO e por influência destes, surge o Cat Swarm Optimization (CSO) desenvolvido por Chu e Tsai em 2007.
- Como o próprio nome indica, tem por base o comportamento dos gatos. Desenvolvido originalmente para resolução de problemas de otimização com espaço de resultados contínuos, este algoritmo acabou por revelar que, com algumas variantes adequadas especificamente às situações em causa, consegue também resolver problemas complexos com propriedades muito próprias.





Problemas de
horários

Processamento
de imagem

Reconhecimento
facial e da fala

**APLICAÇÕES
REAIS**

Otimização de
funções
matemáticas

Problemas de
otimização

Data Mining


Classificação

1.1 COMPORTAMENTO BIOLÓGICO E INTELIGENTE DOS GATOS

- De acordo com a classificação biológica, existem atualmente cerca de 32 espécies diferentes de felinos: leões, tigres, leopardos, gatos, ...
- Padrões de comportamento com semelhanças
 - habilidade para caçar
 - curiosidade sobre objetos que se movam ao seu redor



1.1 COMPORTAMENTO BIOLÓGICO E INTELIGENTE DOS GATOS

- “os gatos têm 7 ou 9 vidas”  VITALIDADE VIOGOROSA
- constante estado de alerta
- extrema atenção a tudo o que acontece ao seu redor
- “modo de caça” no qual perseguem a presa até a conseguirem apanhar
- tempo de descanso > tempo de perseguição de presas

2 ALGORITMO

- Representação da Solução:

- são utilizadas as posições dos gatos no espaço de resultados, de forma a representar uma solução

- Funcionamento geral:

- Inicialmente, define-se o número de gatos a utilizar no algoritmo e aplica-se esse valor ao problema a otimizar;
- Cada gato contém a sua própria posição, composta por M dimensões, velocidades para cada dimensão, valor de fitness (que representa a aptidão do gato segundo a função fitness para o problema), e uma flag para identificar se o gato está no *Seeking Mode* ou no *Tracing Mode*;
- A solução final será a melhor posição de um gato, segundo a função fitness, encontrada pelo algoritmo

2 ALGORITMO

- Parâmetros:

| | |
|-------------------------------------|--|
| Variável de decisão | Posição do gato em cada dimensão |
| Solução | Posição do gato |
| Solução anterior | Posição anterior do gato |
| Nova solução | Nova posição do gato |
| Melhor solução | A posição gato com melhor fitness |
| Função fitness | Distância entre o gato e a presa/objetivo |
| Solução Inicial | Posições aleatórias dos gatos |
| Processo de geração de nova solução | Aplicar os processo de <i>Seeking</i> e <i>Tracing</i> |

2.1 SEEKING MODE E TRACING MODE

- O Cat Swarm Optimization é composto por dois sub-modos que simulam o comportamento dos gatos:
 - **Seeking Mode:** representa o momento em que o gato está a descansar, mas ao mesmo tempo atento ao que o rodeia.
 - **Tracing Mode:** representa o momento em que o gato está a caçar a sua presa.



2.1 SEEKING MODE E TRACING MODE

- Seeking Mode:

- Seeking Memory Pool (SMP) - definir o tamanho de memória de cada gato, isto é, indica o número de soluções vizinhas que este analisa por iteração
- Seeking Range of selected Dimension (SRD) - taxa de alteração para as dimensões selecionadas, impondo limites ao movimento nas mesmas e, portanto, previne que a diferença entre a nova e a antiga posição não saia de um intervalo definido
- Count of Dimensions to Change (CDC) - representa o número de dimensões que vão ser alteradas por iteração
- Self-Position Consideration (SPC) - variável booleana que indica se a atual posição do gato será considerada como movimento para a próxima iteração.

2.1 SEEKING MODE E TRACING MODE

- Seeking Mode:



1º Passo: Fazer j cópias do gato, onde $j = \text{SMP}$. Se o valor do SPC for True, então:

→ $j = (\text{SMP} - 1)$ e consideramos a posição atual do gato como uma das candidatas para a iteração seguinte (uma das cópias é igual à posição atual).

2º Passo: Para cada cópia, de acordo com o valor de CDC, calcular a nova posição usando a seguinte equação:

$$X_{\text{nova}} = (1 \pm \text{SRD} * R) * X_{\text{atual}}$$

X_{nova} nova posição do gato/nova solução

X_{atual} posição atual do gato

R valor aleatório, que varia entre 0 e 1

2.1 SEEKING MODE E TRACING MODE

- Seeking Mode:



3º Passo: Calcular os valores de fitness (FS) para as novas posições. Se todos os valores de FS forem iguais, todas as posições ficam com igual probabilidade de serem selecionadas (dividindo 1 pelo número de SMP). Caso contrário, usar a equação seguinte para calcular a probabilidade de seleção das novas posições:

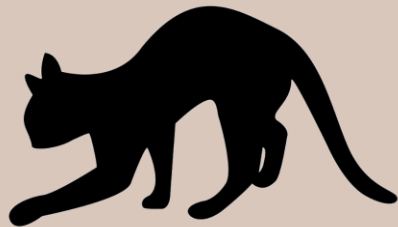
$$P_i = \frac{|FS_i - FS_b|}{|FS_{\max} - FS_{\min}|}, \quad \text{onde } 0 < i < j$$

| | |
|--------------------|--|
| P_i | probabilidade do atual gato _i |
| FS_i | valor de fitness do gato _i |
| FS_{\max} | valor máximo da função de fitness |
| FS_{\min} | valor mínimo da função de fitness |
| $FS_b = FS_{\max}$ | se for para problemas de minimização |
| $FS_b = FS_{\min}$ | se for para problemas de maximização |

4º Passo: Utilizando o método da roleta e de acordo com as probabilidades de seleção (calculadas no passo 3), escolher aleatoriamente a nova posição (solução) para o ~~gato~~_{cat} e atualizamos a posição do mesmo.

2.1 SEEKING MODE E TRACING MODE

- Tracing Mode:



1º Passo: Atualizar as velocidades para cada dimensão ($v_{k,d}$) de acordo com a seguinte equação:

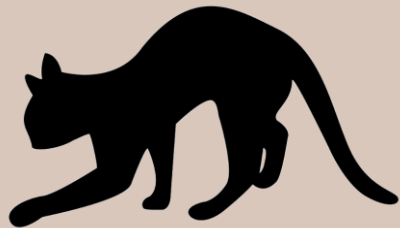
$$v_{k,d} = v_{k,d} + r_1 \times c_1 (X_{\text{best},d} - X_{k,d})$$

| | |
|---------------------|---|
| $v_{k,d}$ | velocidade do gato _k na dimensão _d |
| $X_{\text{best},d}$ | posição do gato com a melhor solução na dimensão _d |
| $X_{k,d}$ | posição do gato _k na dimensão _d |
| c_1 | constante |
| r_1 | valor aleatório entre 0 e 1 |

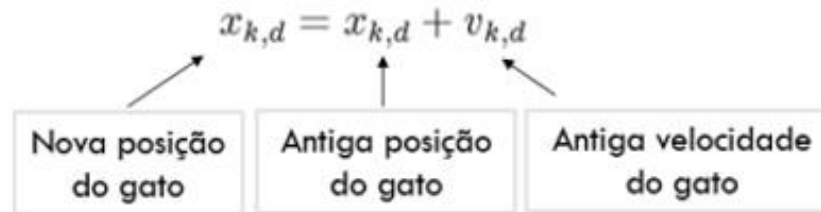
2º Passo: Confirmar se as velocidades não ultrapassam a velocidade máxima. Caso a nova velocidade estiver acima do limite máximo, esta é igualada ao limite.

2.1 SEEKING MODE E TRACING MODE

- Tracing Mode:



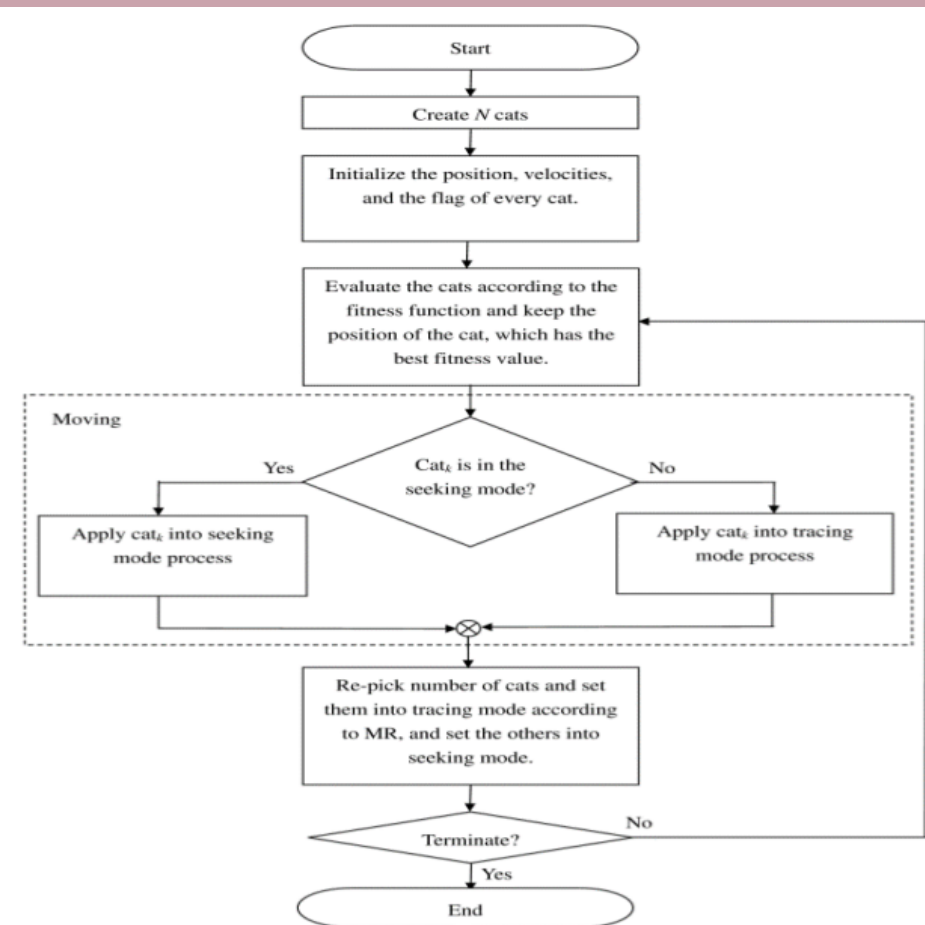
3º Passo: Atualizar a posição do gato_k de acordo com a seguinte equação:

$$x_{k,d} = x_{k,d} + v_{k,d}$$


Nova posição do gato Antiga posição do gato Antiga velocidade do gato

2.2 CAT SWARM OPTIMIZATION

- De forma a combinar e alternar os dois sub-modos no algoritmo, foi definido um rácio de mistura (mixture ratio) MR. Tendo em conta que os gatos passam muito tempo a descansar e pouco a caçar, faz sentido que no algoritmo esteja representado este comportamento, isto é, que o número de gatos em Seeking Mode seja muito superior ao número de gatos em Tracing Mode. Desta forma o valor de MR deve ser um valor muito baixo.



2.2 CAT SWARM OPTIMIZATION

- Vantagens e Desvantagens:
 - O CSO com os seus dois sub-modos de comportamento conjuga o melhor de “dois mundos”.
 - Pesquisa global do espaço de resultados evitando assim que o algoritmo acabe por convergir demasiado cedo para ótimos locais e fique por aí (tracing mode).
 - A performance do algoritmo CSO supera muitos algoritmos tanto em global best solutions encontradas como no número de iterações para o fazer, visto que este converge para o ótimo global com mais facilidade.
 - É mais lento e envolve maiores custos computacionais.

CONCLUSÃO E FASE III

- O Cat Swarm Optimization é um algoritmo que surge fortemente influenciado pelo Ant Colony Optimization (ACO) e o Particle Swarm Optimization (PSO);
- Diversas espécies animais deram origem a algoritmos baseados em Swarm Intelligence considerando as suas características e comportamentos únicos. O Cat Swarm Optimization é um desses algoritmos e tem por base o comportamento dos gatos para solucionar problemas de otimização.
- Pode ser necessário efetuar algumas adaptações ao algoritmo já que o CSO original é especializado na resolução de problemas com espaço de resultados contínuos.

CONCLUSÃO E FASE III

- Os criadores deste algoritmo, efetuaram comparações entre os algoritmos CSO, PSO e uma outra variante deste último, sendo essa comparação realizada com base num benchmark de funções matemáticas.

TABLE 1. The Limitation ranges of dimensions for every test function

| Function Name | Limitation Range |
|-----------------|-------------------------|
| Test Function 1 | $x_d \in [15, 30]$ |
| Test Function 2 | $x_d \in [2.56, 5.12]$ |
| Test Function 3 | $x_d \in [300, 600]$ |
| Test Function 4 | $x_d \in [-30, 30]$ |
| Test Function 5 | $x_d \in [-5.12, 5.12]$ |
| Test Function 6 | $x_d \in [-100, 100]$ |

TABLE 2. Parameter settings for CSO

| Parameter | Value or Range |
|------------|----------------|
| <i>MP</i> | 5 |
| <i>SRD</i> | 20% |
| <i>CDC</i> | 80% |
| <i>MR</i> | 2% |
| c_1 | 2.0 |
| r_1 | [0, 1] |

TABLE 3. Parameter settings for PSO and PSO with WF

| Parameter | Value or Range |
|----------------|----------------|
| Initial Weight | 0.9 |
| Final Weight | 0.4 |
| c_1 | 2.0 |
| c_2 | 2.0 |
| r_1 | [0, 1] |
| r_2 | [0, 1] |

TABLE 4. Maximum velocities for PSO and PSO with WF

| Function Name | Maximum Velocity |
|-----------------|------------------|
| Test Function 1 | 100.0 |
| Test Function 2 | 10.0 |
| Test Function 3 | 600.0 |
| Test Function 4 | 100.0 |
| Test Function 5 | 600.0 |
| Test Function 6 | 10.0 |

CONCLUSÃO E FASE III

- Os resultados dos vários testes que fizeram, não lhes deixaram margem para dúvidas:

- a performance do algoritmo CSO supera largamente os algoritmos PSO-type, tanto em global best solutions encontradas, como no número de iterações para o fazer, visto que este converge para o ótimo global com mais facilidade.

- é de notar que este é mais lento e envolve um maior custo computacional comparativamente com os outros algoritmos em estudo.

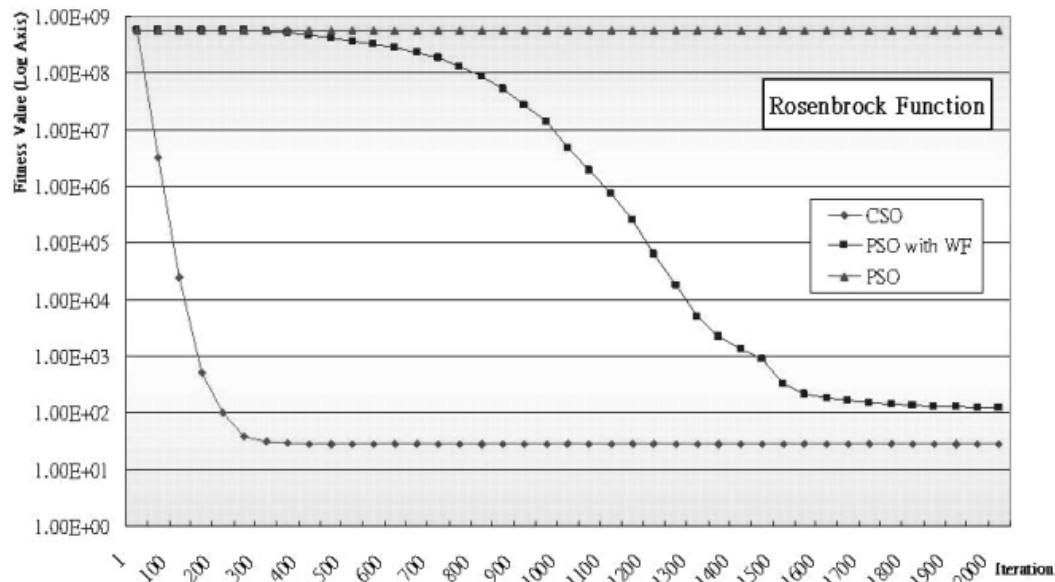


FIGURE 2. The experimental result of test function 1

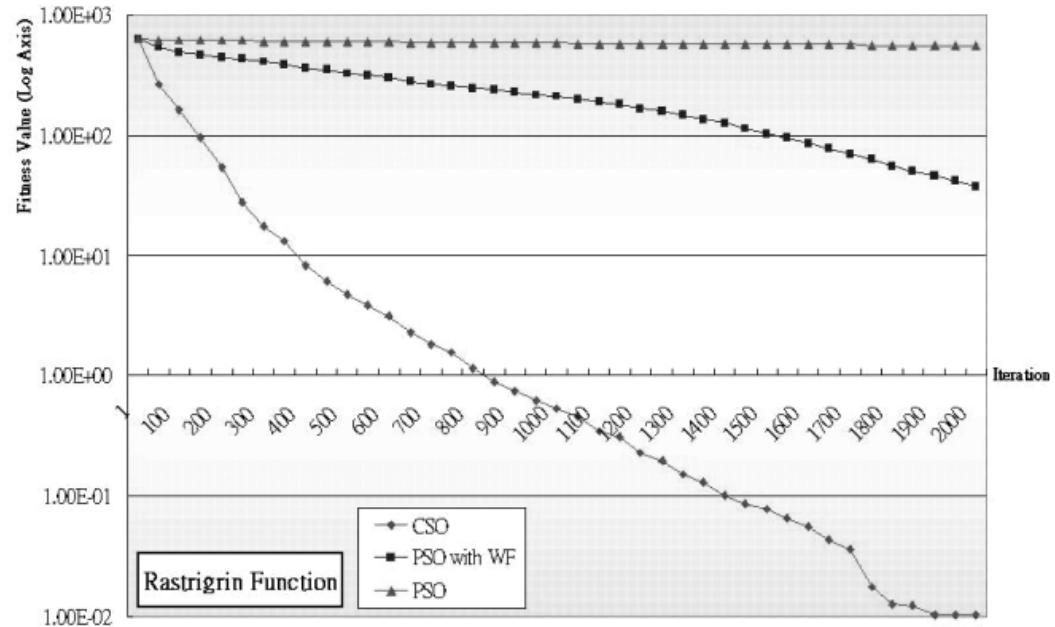


FIGURE 3. The experimental result of test function 2

CONCLUSÃO E FASE III

- Na fase seguinte deste projeto pretendemos, nós mesmos, comprovar estes factos conclusivos tendo por base as implementações disponíveis no repositório GitHub.
- Pela definição de alguns hyper-parâmetros do modelo segue-se a procura pela melhor configuração. No final da fase III do projeto devemos ser capazes de analisar e avaliar diferentes soluções e estabelecer uma análise comparativa e crítica fundamentada.



REFERÊNCIAS BIBLIOGRÁFICAS

Bahrami M., Bozorg-Haddad O. e Chu X., Cat Swarm Optimization (CSO) Algorithm, Capítulo 2. Disponível online em:

https://www.researchgate.net/publication/318128829_Chapter_2_Cat_Swarm_Optimization_CS_O_Algorithm

Chu SC., Tsai P., Pan JS. (2006) Cat Swarm Optimization. In: Yang Q., Webb G. (eds) PRICAI 2006: Trends in Artificial Intelligence. PRICAI 2006. Lecture Notes in Computer Science, vol 4099. Springer, Berlin, Heidelberg.

https://doi.org/10.1007/978-3-540-36668-3_94

Chu SC. e Tsai P. (2007), Computational intelligence based on the behavior of cats. Disponível online em:

https://www.researchgate.net/publication/228721750_Computational_intelligence_based_on_the_behavior_of_cats