

Instituto Superior de Engenharia de Coimbra

Licenciatura Eng. Informática

3ºAno/1ºSemestre

2020/2021

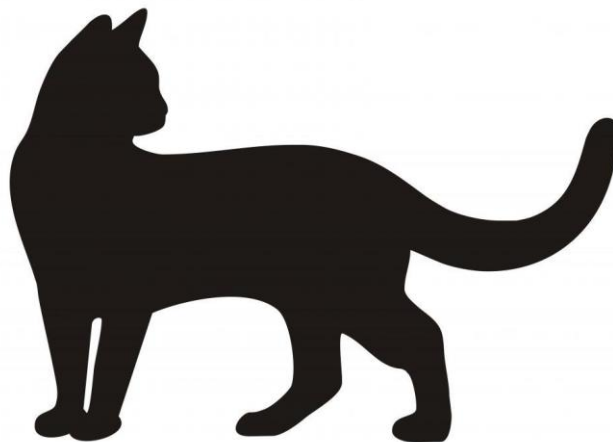


# INTELIGÊNCIA COMPUTACIONAL

---

## CAT SWARM OPTIMIZATION

Projeto – Fase II – Seminário



Carolina Ferreira nº 2018018459

Eduardo Pina nº2017010181

# 1 ÍNDICE

---

2	Introdução .....	3
2.1	Comportamento biológico e inteligente dos gatos .....	3
3	Algoritmo .....	4
3.1	Representação da solução.....	4
3.2	Funcionamento geral .....	5
3.3	Critério de paragem.....	5
3.4	Parâmetros.....	5
3.5	Seeking Mode e Tracing Mode .....	5
3.5.1	Seeking Mode.....	6
3.5.2	Tracing Mode .....	7
3.6	Cat Swarm Optimization.....	8
3.6.1	Algoritmo .....	8
3.6.2	Vantagens e Desvantagens .....	8
4	Conclusão e Fase III.....	9
5	Referências Bibliográficas .....	10

## 2 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: o Genetic Algorithm (GA), o Ant Colony Optimization (ACO) e o Particle Swarm Optimization (PSO). O Algoritmo Genético (GA), na qualidade de algoritmo evolucionário, usa uma população de soluções, e ao longo de várias gerações vai aplicando operadores de seleção, crossover, mutação e de reprodução de forma a solucionar os problemas aos quais é aplicado. Os algoritmos ACO e PSO, são baseados em Swarm Intelligence (SI). Neste sistema de modelação, uma população (ou enxame) de organismos, tais como formigas (ACO), abelhas, peixes ou aves (PSO) interagem entre si e com o ambiente, partilhando informação de forma a solucionar um problema de otimização. É importante referir que estes algoritmos são fortemente influenciados pelo tipo de enxame escolhido, ou seja, pelas propriedades únicas que definem os animais no seu nível e biológico e também pelo seu comportamento inteligente.

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.

O CSO e as suas variantes podem então ser aplicados em diversos problemas e são exemplos reais os problemas de horários, processamento de imagem, reconhecimento facial e da fala, otimização de funções matemáticas, problemas de otimização, data mining, classificação, entre muitos outros.

### 2.1 COMPORTAMENTO BIOLÓGICO E INTELIGENTE DOS GATOS

Sendo o CSO inspirado no comportamento dos gatos, é importante fazer um estudo mais aprofundado para entender melhor o âmagos e a essência deste algoritmo.

De acordo com a classificação biológica, existem atualmente cerca de 32 espécies diferentes de felinos: leões, tigres, leopardos, gatos, etc. Embora muitos destes tenham habitats diferentes, os felinos apresentam padrões de comportamento com semelhanças entre si como, por exemplo, a sua habilidade para caçar. Esta habilidade é adquirida através de treino, não é algo que todos os felinos saibam fazer a partir do momento em que nascem. É uma capacidade não inata, apesar da aptidão natural para isso, que tem de ser adquirida através do treino e experiência. Para os felinos selvagens, a destreza para caçar garante-lhes alimento, e, portanto, é vital para a sua sobrevivência. No caso dos felinos domésticos, mais precisamente, os gatos domésticos (no qual o CSO é baseado), embora muitas das vezes não necessitem de caçar para sobreviver, demonstram a também a destreza natural para a caça e curiosidade sobre objetos que se movam ao seu redor.

Algumas culturas acreditam que “os gatos têm 7 ou 9 vidas”, crença que não deve ser levada no seu sentido literal. A afirmação é uma referência ao facto destes felinos possuírem uma vitalidade vigorosa. É frequente ouvir um gato ronronar e esse som é interpretado como um estado de felicidade, perigo ou doença. Apesar de os gatos aparentarem passar grande parte do seu tempo a descansar, a verdade é que os gatos estão em constante estado de alerta com extrema atenção a tudo o que acontece ao seu redor e quando notam alguma presa ou algo que os atraia, passam ao seu “modo de caça” no qual perseguem a presa até a conseguirem apanhar. Tendo em conta o tempo que um gato dedica ao seu descanso, o tempo que estes passam a perseguir as suas presas é muito menor, e, desta forma, os gatos conseguem economizar bastante a sua energia.

## 3 ALGORITMO

---

### 3.1 REPRESENTAÇÃO DA SOLUÇÃO

Todos os algoritmos de otimização necessitam de uma forma de representação das soluções para os problemas em mãos e que pretendem solucionar (por exemplo: representando por um array binário, um 1 na posição [i] do mesmo representa que o objeto está dentro de algo, que existe ou que pertence ali e, por sua vez, o valor 0 indica o contrário). No Cat Swarm Optimization (CSO), são utilizadas as posições dos gatos no espaço de resultados, de forma a representar uma solução.

### 3.2 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. O CSO guarda a posição do melhor gato até chegar ao final das iterações, tendo estas um limite pré-definido.

### 3.3 CRITÉRIO DE PARAGEM

Definir um bom critério de paragem é uma regra importante para garantir uma boa convergência do algoritmo. O número de iterações, a quantidade de melhoria e o tempo de execução são critérios comuns no algoritmo CSO e nas suas variantes.

### 3.4 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>

### 3.5 SEEKING MODE E TRACING MODE

O Cat Swarm Optimization é composto por dois sub-modos que simulam o comportamento dos gatos. O Seeking Mode, representa o momento em que o gato está a descansar, mas ao mesmo

tempo atento ao que o rodeia. No caso do Tracing Mode, este representa o momento em que o gato está a caçar a sua presa.

### 3.5.1 Seeking Mode

Antes de analisar o algoritmo para este modo comportamental é importante referir que o Seeking Mode é constituído por quatro fatores essenciais: o Seeking Memory Pool (SMP), Seeking Range of selected Dimension (SRD), Count of Dimensions to Change (CDC) e Self-Positon Consideration (SPC). Em primeiro lugar, o SMP é utilizado para definir o tamanho de memória de cada gato, isto é, indica o número de soluções vizinhas que este analisa por iteração. O SRD simboliza a taxa de alteração para as dimensões seleccionadas, 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. No caso do CDC, este representa o número de dimensões que vão ser alteradas por iteração e, por último, o SPC é uma variável booleana que indica se a atual posição do gato será considerada como movimento para a próxima iteração.

**1º Passo:** Fazer  $j$  cópias do gato<sub>k</sub>, 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:

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

$X_{\text{nova}}$  nova posição do gato/nova solução

$X_{\text{atual}}$  posição atual do gato

$R$  valor aleatório, que varia entre 0 e 1

**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 seleccionadas (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 <sub>i</sub>
$FS_i$	valor de fitness do gato <sub>i</sub>
$FS_{\max}$	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<sub>k</sub> e atualizamos a posição do mesmo.

### 3.5.2 Tracing Mode

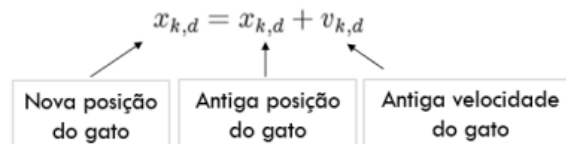
**1º Passo:** Atualizar as velocidades para cada dimensão ( $v_{k,d}$ ) de acordo com a seguinte equação:

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

$v_{k,d}$	velocidade do gato <sub>k</sub> na dimensão <sub>d</sub>
$X_{\text{best},d}$	posição do gato com a melhor solução na dimensão <sub>d</sub>
$X_{k,d}$	posição do gato <sub>k</sub> na dimensão <sub>d</sub>
$c_1$	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.

**3º Passo:** Atualizar a posição do gato<sub>k</sub> de acordo com a seguinte equação:



## 3.6 CAT SWARM OPTIMIZATION

Como mencionado acima, o Cat Swarm Optimization tem dois sub-modos, Seeking Mode e Tracing Mode. De forma a combinar e alternar estes dois 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.

### 3.6.1 Algoritmo

**1º Passo:** Criar N gatos;

**2º Passo:** Espalhar aleatoriamente os gatos nas M dimensões do espaço de soluções e, aleatoriamente também selecionar valores para a velocidade de cada gato, que estejam dentro do limite máximo pré-definido. Depois, ao acaso, escolher um determinado número de gatos para passar para Tracing Mode, de acordo com o MR, e colocar os restantes em Seeking Mode;

**3º Passo:** Aplicar a cada gato (posição) a função fitness de forma a calcular a aptidão do mesmo e guardar em memória o gato com melhor fitness (**nota: apenas precisaríamos de armazenar a posição do mesmo, visto que esta representa a melhor solução até então**);

**4º Passo:** Movimentar todos os gatos conforme o seu sub-modo (aplicar o processo de movimento conforme o gato<sub>k</sub> se encontre em Tracing Mode ou em Seeking Mode);

**5º Passo:** Escolher novamente o número de gatos para colocar, segundo o MR, em Tracing Mode, e aplicar o Seeking Mode aos restantes;

**6º Passo:** Verificar o critério de paragem definido, e, se a condição for satisfeita, parar o algoritmo. Caso contrário, recomeçar do 3º Passo.

### 3.6.2 Vantagens e Desvantagens

O CSO com os seus dois sub-modos de comportamento conjuga o melhor de “dois mundos”. Cada gato que se encontra em Seeking Mode realiza uma pesquisa local no espaço de resultados (tal como um trepa-colinas ou qualquer outro método de pesquisa local), mas, no sub-modo Tracing Mode, este realiza uma pesquisa global do espaço de resultados, evitando assim que o algoritmo acabe por convergir demasiado cedo para ótimos locais e fique por aí.



Segundo os criadores deste algoritmo que, obviamente, executaram inúmeras baterias de testes para avaliar o desempenho do algoritmo por eles desenvolvido, 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. No entanto, notaram que este é mais lento e envolve maiores custos computacionais.

## 4 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). O ACO é um algoritmo usado para encontrar caminhos ótimos baseado no comportamento das formigas em busca de alimento. No início, as formigas vagueiam aleatoriamente e quando encontram uma fonte de alimento, caminham de volta para a colônia deixando "marcadores" (feromonas) que mostram que o caminho conduz a alimento. O PSO, algoritmo evolucionário, tem como inspiração a natureza biológica, o comportamento social e os movimentos dinâmicos com comunicação de insetos, peixes e pássaros.

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. Existe uma grande diversidade de implementações e de variações dos algoritmos para a resolução de um leque de problemas de otimização, nos mais diversos campos científicos. No entanto, dependendo do problema e das suas características que se pretende otimizar pode ser necessário efetuar algumas adaptações ao algoritmo já que o CSO original é especializado na resolução de problemas com espaço de resultados contínuos.

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. 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. No entanto, é de notar que este é mais lento e envolve um maior custo computacional comparativamente com os outros algoritmos em estudo.

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.

## 5 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](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](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](https://www.researchgate.net/publication/228721750_Computational_intelligence_based_on_the_behavior_of_cats)

