

# Computação Natural - Particle Swarm Optimization Benchmark

Diogo F. Braga

University of Minho

Department of Informatics

4710-057 Braga, Portugal

Email: a82547@alunos.uminho.pt

**Resumo.** Uma meta-heurística define-se como um método que tenta resolver de forma genérica problemas de otimização e que, normalmente, é aplicada a problemas para os quais não se conhece um algoritmo eficiente. Neste sentido, surge o Particle Swarm Optimization, um algoritmo com base na Computação Evolucionária que proporciona uma solução para estes problemas.

## 1 Introdução

O Particle Swarm Optimization (PSO) apresenta-se como um algoritmo baseado em sistemas sociais, como o comportamento em grupo das aves, e normalmente é utilizado para otimização de problemas. Trata-se de um método computacional que otimiza um problema, iterativamente, ao tentar melhorar a solução candidata com respeito a uma dada medida de qualidade. Este resolve um problema ao ter uma população de soluções candidatas, denominadas partículas, e movendo-as no espaço de pesquisa de acordo com fórmulas matemáticas sobre a posição e a velocidade da partícula.

Neste documento é realizado um *benchmark* com alteração dos principais parâmetros variáveis das fórmulas matemáticas associadas ao algoritmo, em específico: número de partículas; constante individual; constante de grupo; inércia. O *benchmark* é realizado em relação ao seguinte exercício, cuja ideia é minimizar o valor resultante da seguinte função.

$$\begin{aligned} f(x) &= \sum_{i=1}^N (x_i - a_i)^2 \\ N &= 30 \\ -100 &\leq x_i \leq 100 \\ -80 &\leq a_i \leq 80 \end{aligned}$$

Where  $a_i$  is selected previously

Fig. 1. Exercício utilizado para realização do benchmark.

## 2 Cooperação e coordenação

Quando existem vários agentes (partículas) num ambiente, cada agente enfrenta um problema de planeamento conjunto, no qual tenta atingir seus próprios objetivos com a ajuda dos outros. Podemos obter uma simulação razoável de um rebanho se cada agente de aves (às vezes chamado de *boid*) observar as posições de seus vizinhos mais próximos e escolher a posição e a velocidade que maximizem a soma ponderada dessas duas componentes. Se todos os *boids* executarem essa política, o rebanho exibirá o comportamento emergente de voar como um corpo pseudo-rígido com densidade aproximadamente constante que não se dispersa ao longo do tempo e que ocasionalmente faz movimentos repentinos. Este conceito aplica-se no problema referenciado neste documento, sendo que este exemplo em causa de comportamento multiagente cooperativo encontra-se representado no comportamento de *flock* de aves.

## 3 Alteração do número de partículas

De forma a poder realizar uma avaliação fidedigna a este parâmetro e uma boa comparação entre as alterações, foram mantidos estáveis os valores de entrada do exercício. Desta forma, com  $x = 50$  e  $a = 10$ , os resultados foram os apresentados na figura 2. Desse gráfico é possível concluir que quanto mais partículas melhores resultados, o que era esperado. No entanto, notar que a utilização de 1000 partículas talvez já seja um esforço computacional exagerado pois, apesar de realizar uma aproximação mais próxima dos melhores resultados em relação às 100 partículas, esta só coincide com a zona de estabilização da solução ótima no mesmo momento que o caso das 100 partículas. Desta forma, as seguintes comparações serão feitas com número de partículas igual a 100, que atinge soluções com valores por volta dos 0.0001.

## 4 Alteração da constante individual

De forma a notar-se realmente a influência que esta constante produz na procura da solução ótima, o valor foi colocado a nulo. Isto significa que só a componente social

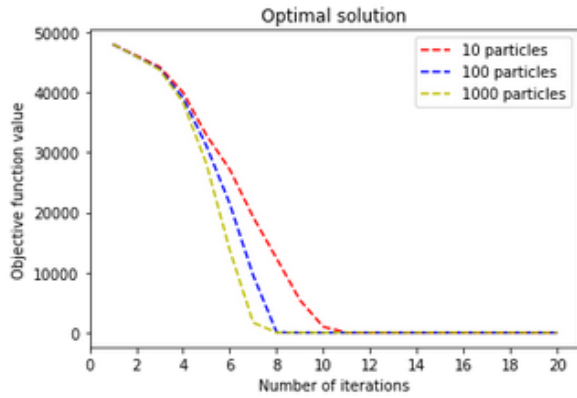


Fig. 2. Resultados obtidos com variação no número de partículas.

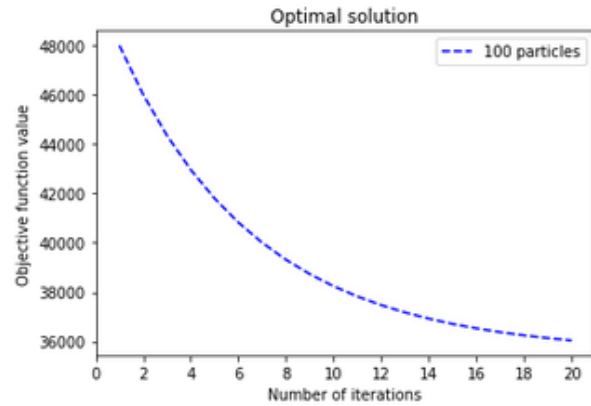


Fig. 3. Resultados obtidos com constante global nula.

tem influência nas decisões tomadas, e que consequentemente, a *exploitation* atinge valores máximos e a *exploration* atinge valores mínimos. Tal acontece porque, com estes valores, não existem influências individuais de cada partícula, levando assim a que todas sigam o mesmo caminho. Esta solução pode tornar-se um problema devido à falta de exploração que, consequentemente, pode levar a solução a atingir um mínimo local mais cedo que o desejável, ficando assim mais longe do mínimo global.

No entanto, para o problema em causa, é sempre atingida uma boa solução na ordem dos valores atingidos inicialmente, e ainda que a solução não seja a ótima, realizar esta abordagem parece não se tornar muito desvantajosa. Ainda assim, é desaconselhável pelas razões anteriormente apresentadas.

## 5 Alteração da constante global

Tal como na secção anterior, esta constante foi também colocada a nulo. Isto significa que só a componente individual tem influência nas decisões tomadas, e que consequentemente, a *exploration* atinge valores máximos e a *exploitation* atinge valores mínimos. Tal acontece porque, com estes valores, não existe planeamento entre todas as partículas, levando assim a que cada uma siga um próprio caminho. Esta solução parece ser um grande problema devido à falta de "comunicação" entre os vários agentes que, desta forma, não vão coincidir na solução ótima. Com esses valores, os resultados foram os apresentados na figura 3. Neste gráfico é possível constatar um resultado falhado, com valores ótimos extremamente longes dos atingidos com esta constante positiva. Apesar de se encontrar em descida, este mostra ser imensamente ineficiente, mesmo na 20ª iteração.

## 6 Alteração do valor da inércia

Esta variável define a influência que cada iteração possui nos cálculos de procura pela solução ótima. Com normalmente valores compreendidos entre o 0.4 e o 0.9, a influência desta variável não é tão grande como as anteriores, no entanto deve ser utilizada de forma eficiente. Os valores elevados privilegiam a *exploration*, pois fazem com que cada de-

cisão seja o mais abrangente possível, enquanto que valores mais baixos privilegiam a *exploitation*, pois a procura é mais cuidadosa e lenta. Qualquer uma das opções nos seus extremos se apresentam como más opções, devendo assim ser combinadas para proporcionar o melhor resultado ao problema em causa. Uma possível solução que coincide as duas técnicas é inicializar com valores mais elevados, de forma a providenciar alguma exploração inicial, e depois, com o passar das iterações, diminuir esse valor de forma a que seja diminuída a exploração e se tente convergir para uma solução ótima.

## 7 Conclusão

Deste documento e pesquisas relacionadas é importante, inicialmente, constatar que estes algoritmos são úteis principalmente em problemas que não existem soluções mais específicas, neste caso, meta-heurísticas. Importante, também, referir que com este algoritmo não se garante uma solução ótima, mas sim uma solução com uma qualidade aceitável num tempo muito curto, sendo esta uma das principais vantagens da sua utilização.

Importante referir os valores de entrada, que são algo que influencia consideravelmente o desempenho do algoritmo, pois este pode iniciar numa solução bastante longe da ótima e até mesmo numa zona com bastantes mínimos locais, levando a um rápido convergir das partículas e consequente pouca eficácia.

Quanto aos fatores de *exploitation* e *exploration*, é muito importante ter capacidade de balancear os dois, pois são estas duas variáveis que definem principalmente o rumo que a pesquisa toma. Dentro desta questão, fazer referência à solução apresentada na secção 6, em que existe uma variação gradual entre esses dois fatores, com início nos valores elevados de *exploration*.

Para finalizar, referir a extrema importância da lógica de grupo que existe neste algoritmo, com uma coordenação constante entre as partículas, que faz com que sejam abordadas inúmeras soluções para o problema em causa, encontrando sempre uma solução próxima da ótima. Sem cooperação e coordenação entre as partículas, o algoritmo não possuiria qualquer efeito.