Laboratório 4 - Otimização com Métodos Baseados em População

Marcelo Buga Martins da Silva

CT-213 - Professor Marcos Ricardo Omena de Albuquerque Máximo

10/04/2021

1 Implementação do PSO

Nesse laboratório foi implementado o algoritmo *Particle Swarm Optmization* (PSO) para a resolução de problemas de otimização. A maior particularidade da implementação desse método de otimização nesse laboratório foram a utilização de duas classes, *Particle* e *ParticleSwarmOptimization*.

O construtor de *Particle* inicializa posições e velocidade aleatórias da partícula dentro de um limite especificado, assim como inicializa variáveis que guardarão as coordenadas e o custo da melhor posição histórica da partida.

Por sua vez, o construtor de *ParticleSwarmOptimization* inicializa um número determinado de partículas, assim como inicializa variáveis que guardam os dados da melhor iteração da geração, assim como a melhor iteração global. Por fim, optou-se por inicializar um contador nessa função para auxiliar as demais funções a encontrarem uma determinada partícula, sendo também utilizada para atualizar os dados da melhor iteração e do melhor global depois de passar por uma geração. Dentro dessa classe, destacam-se os métodos *notify_evaluation* e *advance_generation*. Ambos foram utilizados para atualizar os valores de cada partícula, aplicando, de fato, o algoritmo do PSO.

2 Teste da implementação do PSO

Para testar a implementação, partiu-se de um problema muito simples: encontrar o valor que maximiza a Equação 1:

$$f(x) = -\left((x(0)-1)^2 + ((x(1)-2)^2 + ((x(2)-3)^2\right) \tag{1}$$

Tal equação possui máximo global (trivial) em $(1, 2, 3)$. Utilizando o algoritmo implementado, obeteve-se resultado muito satisfatório, com uma convergência bem rápida dos valores. Foi encontrado o valor exato para os parâmetros x(0), x(1) e x(2), como mostram as Figuras 1, 2 e 3:

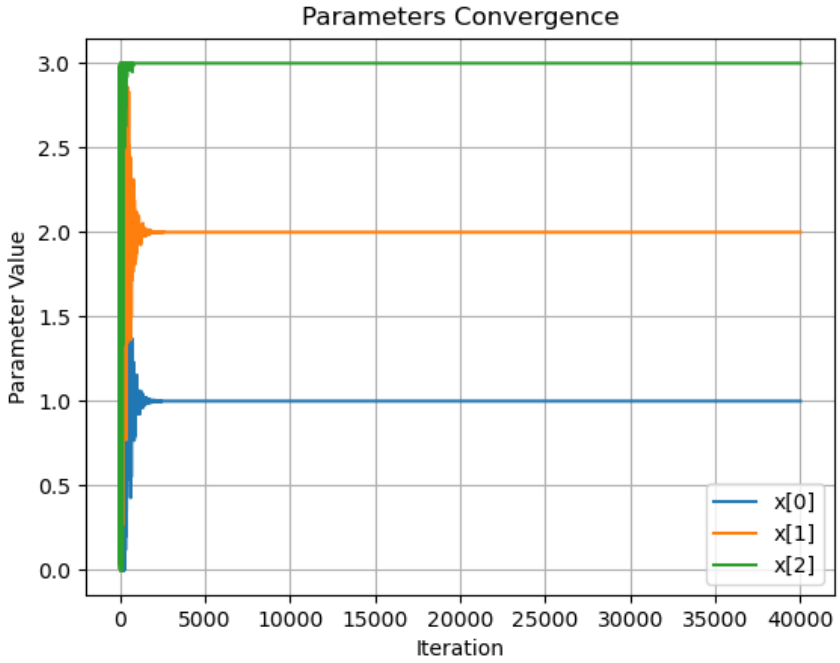

Figura 1: Convergência dos parâmetros para o caso teste

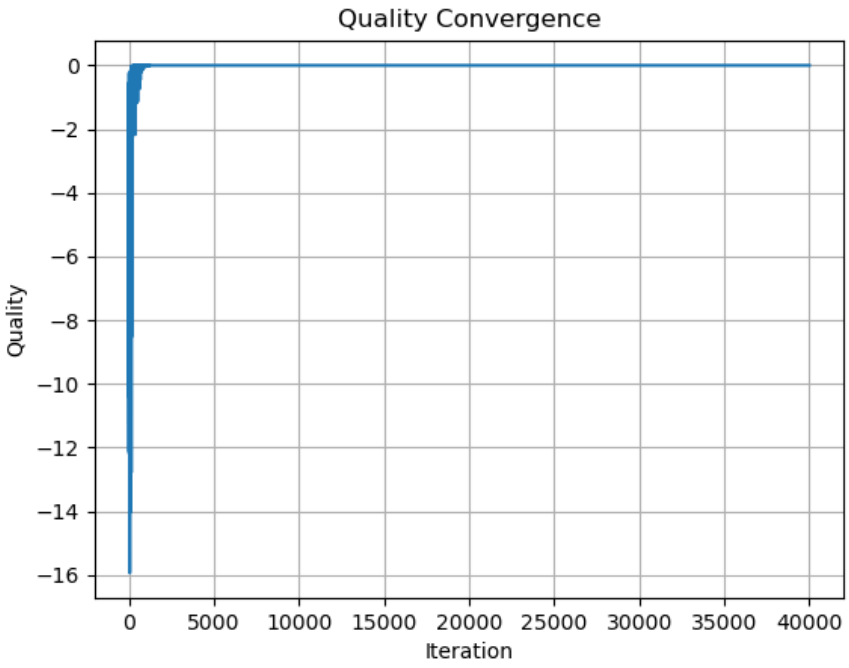

Figura 2: Convergência da qualidade geral das partículas

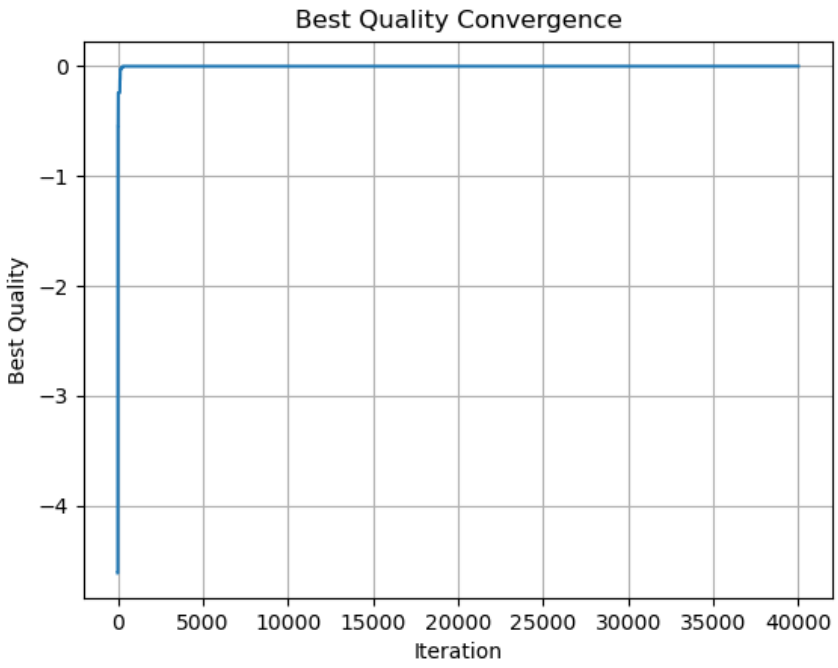

Figura 3: Convergência da melhor qualidade das partículas

Com a implementação tendo se provada muito boa para o caso teste, decidiu-se então utilizá-la para o

problema muito mais complicado proposto: a otimização dos parâmetros de controle para um robô *Line Follower*.

3 PSO para otimização de parâmetros de controle

Dado que o algoritmo do PSO já havia sido implementado e testado, o maior desafio nessa parte do laboratório foi criar uma função de recompensa para o bom comportamento do robô. Para tal recompensa, foi utilizada a Equação 2:

$$f(x) = \sum_{k}^{N} \left(v_k \left\langle r_k, t_k \right\rangle - w \left| e_k \right| \right) \tag{2}$$

Nessa equação, N é a duração do episódio de treinamento em passos de tempo (*time steps*) e k é o instante, de tal forma que v_k é a velocidade linear, r_k é um vetor (bidimensional) unitário que aponta na direção do robô no instante k, t_k é o vetor tangente à atual posição no caminho no instante k,$|e_k|$ é o módulo do erro em relação à linha e w é um peso para fazer um compromisso entre se manter no centro da linha e seguir o caminho rapidamente.

Nessa primeira implementação, foi utilizado $w = 0,5$ e convencionado que $e_k = 0,06$ quando o robô está fora da linha. Infelizmente, essa implementação não teve resultados nada positivos, com o robô convergindo para seguir reto depois da primeira curva.

Depois de várias tentativas, a que obteve o melhor resultado foi a que teve as seguinte considerações:

$w = 0,6$

$e_k = 0,08$ e $\langle r_k, t_k \rangle = -0,02$ quando o robô está fora da linha.

Utilizou-se desse artifício para punir mais o robô por sair da linha. Em geral o resultado final foi bem satisfatório, embora a curva mais difícil do final do percurso não tenha sido tão bem realizada. Foram utilizadas mais de 15.000 iterações para chegar aos resultados das Figuras 4, 5 e 6.

Os melhores parâmetros encontrados foram:

1. *Linear Speed*: 0,745178
2. Kp: 89,101469
3. Ki: 826,669672
4. Kd: 16,313806

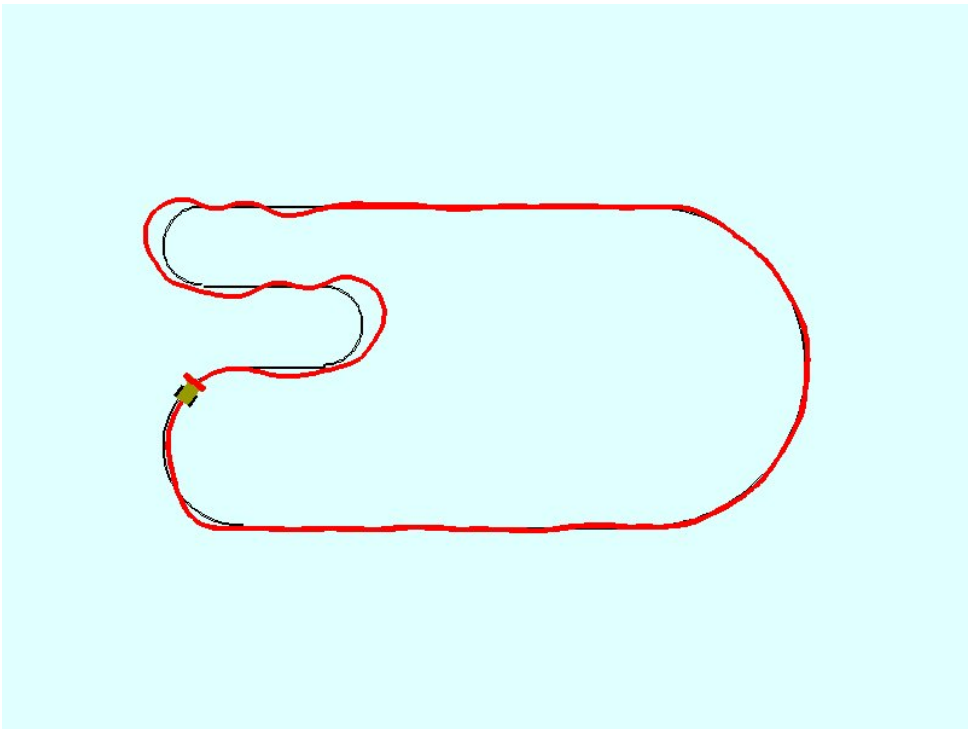

Figura 4: Melhor solução para o *Line Follower*

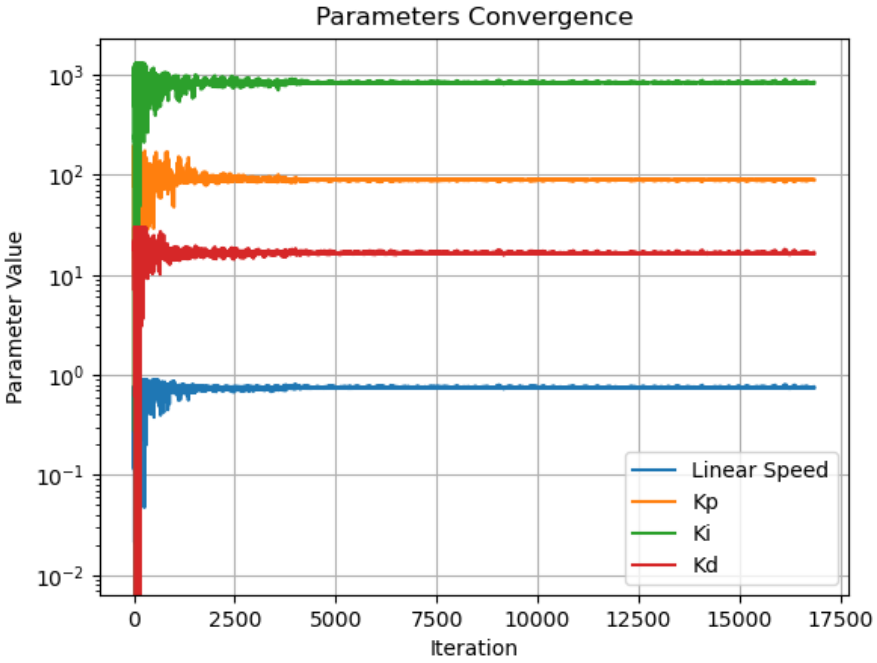

Figura 5: Convergência dos melhores valores dos parâmetros de controle

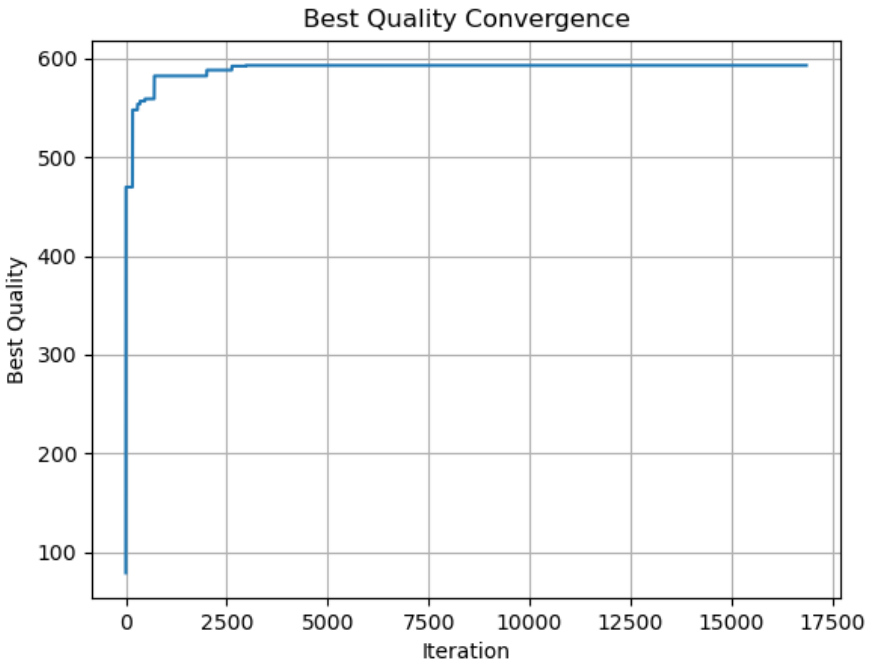

Figura 6: Convergência da melhor qualidade das partículas

Em comparação com a solução de exemplo do professor (Figura 7, percebe-se que o caminho do carrinho desviou um pouco mais da linha, embora (provavelmente, dado que o código base pega um dos instantes finais para o *screenshot*) tenha conseguido ir mais longe. Como a função de recompensa considera a velocidade, é provável que, para a função de recompensa utilizada, a solução aqui apresentada seja "melhor"do que a de exemplo. Para atingir algo mais próximo da Figura 7, seria, então, provavelmente necessário recompensar mais por seguir a linha e menos pela velocidade.

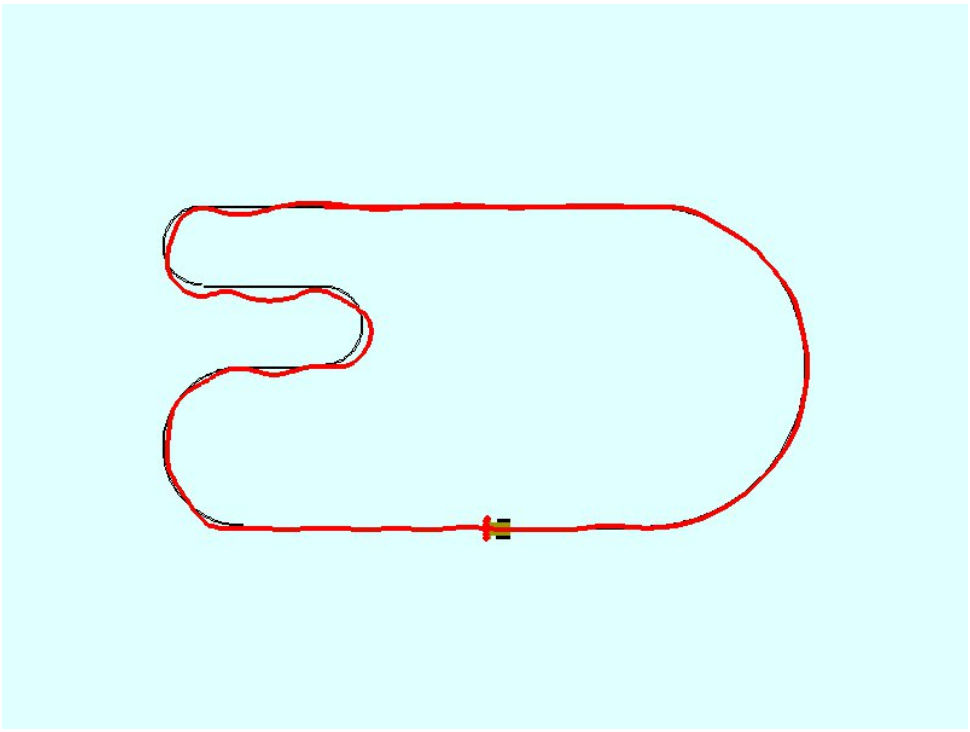

Figura 7: Exemplo de resultado da otimização dado pelo professor

Por fim, algo interessante foi o comportamento da convergência geral das partículas, explicitado pelas Figuras 8 e 9. Percebe-se que muitas partículas ficaram presas em algum máximo local, o que impediu a convergência de todas as partículas para um mesmo valor, justificando o comportamento "caótico"dos gráficos.

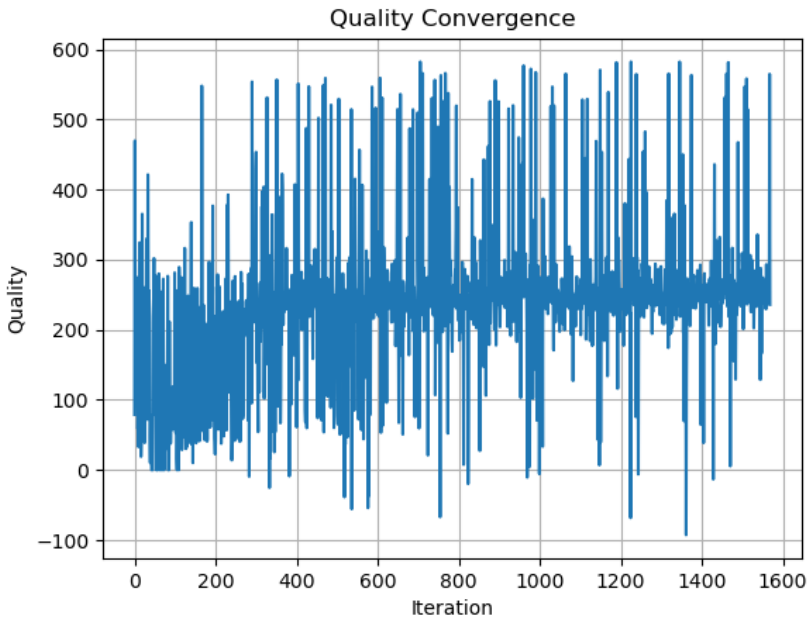

Figura 8: Qualidade das partículas nas primeiras 1500 iterações

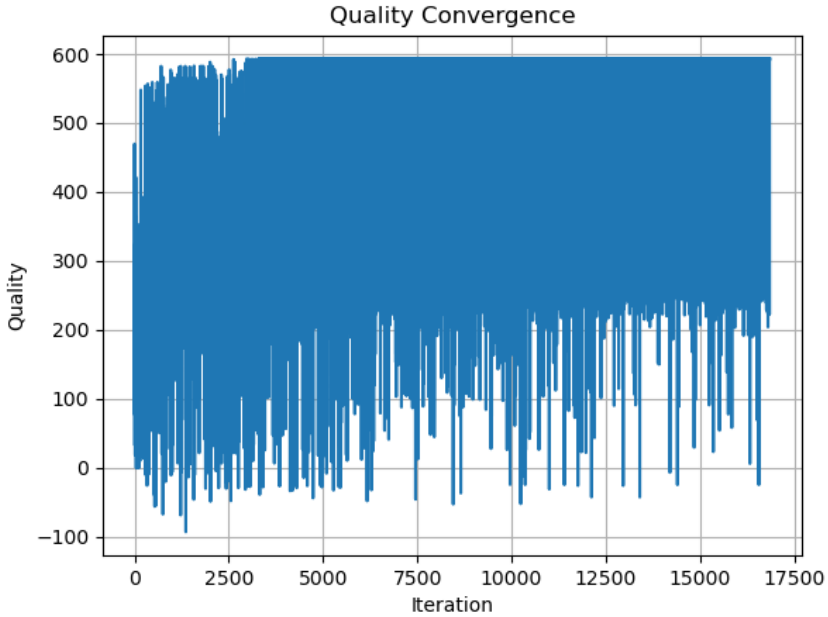

Figura 9: Qualidade das partículas após todas as iterações