

Otimização com Métodos Baseados em População

Samara Ribeiro Silva

Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcos Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 10 de abril de 2021.

O algoritmo de otimização PSO (Particle Swarm Optimization) consiste em armazenar a posição e o custo de um conjunto de partículas e a medida que as iterações de atualização das partículas forem ocorrendo os dados da melhor iteração de cada partícula e a melhor partícula global, também, são armazenadas.

As atualizações ocorrem utilizando os seguintes cálculos:

$$v_i = \omega v_i + \varphi_p r_p (b_i - x_i) + \varphi_g r_g (b_g - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

- ω = peso inercial
- φ_p = parâmetro cognitivo
- φ_g = parâmetro social
- $r_p, r_g \sim U([0, 1])$
- b_i = melhor posição de uma partícula
- b_g = melhor posição considerando todas as partículas

Os parâmetros ω , φ_p , φ_g , r_p e r_g são hiperparâmetros e devem ser inseridos de acordo com a heurística de cada problema.

As posições iniciais de cada partícula foram geradas randomicamente considerando uma distribuição uniforme entre os limites inferior e superior para cada dimensão. Também foi realizado um ajuste para que as partículas não ultrapassem os limites do campo de interesse pré determinados. A velocidade inicial também foi gerada $v_i \sim U(-(\lim sup - \lim inf), (\lim sup - \lim inf))$ e foi realizado quando a posição for alterada para não ultrapassar a fronteira.

Observe na Figura 1, 2 e 3 que o algoritmo implementado obteve sucesso no teste realizado.

Figura 1: Best Quality Converge do teste do PSO.

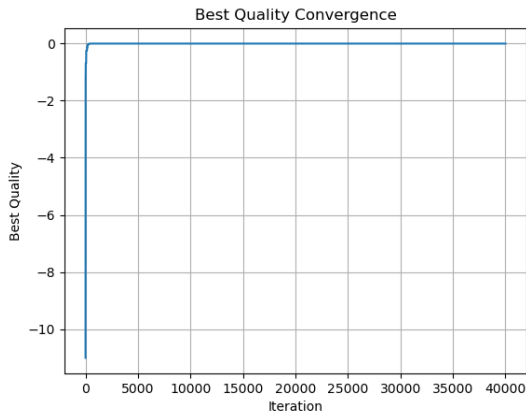


Figura 2: Gráfico da convergência de parâmetros do teste do PSO.

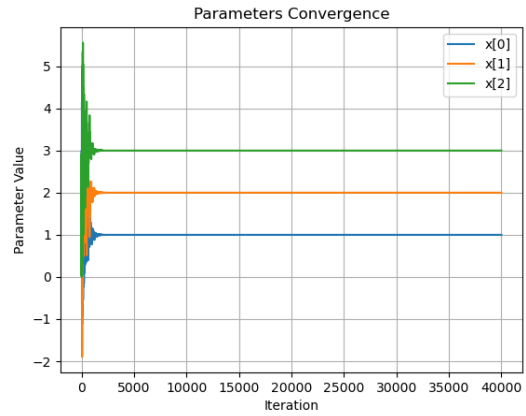
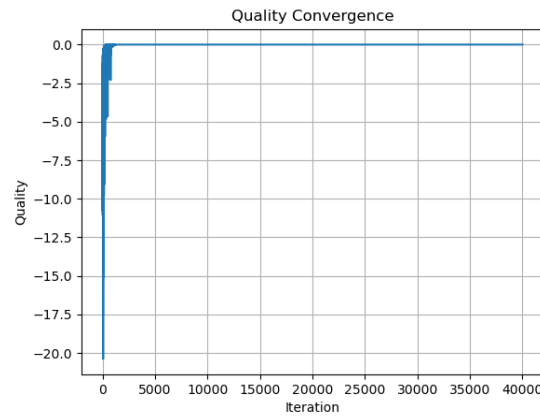


Figura 3: Quality convergence do teste do PSO



Para a simulação foi acrescentada uma função de recompensa para incentivar a melhora dos resultados onde quanto maior o erro (linha detectada) menor é a recompensa. Quando a linha não é detectada considera o erro como 1, pois a função de cálculo de erro analisa apenas o erro enquanto a linha está sendo detectada e retorna valor nula para linha não detectada.

A função recompensa considerada foi:

$$reward_k = v_k \cdot dot(r_k, t_k) - w |e_k| \quad (3)$$

- v_k = velocidade linear
- $dot(r_k, t_k)$ = produto interno entre r_k e t_k
- w = peso para fazer um compromisso entre se manter no centro da linha e seguir o caminho rapidamente.

- $|e_k|$ = valor absoluto do erro em relação a linha

Observe nas figuras 4, 5, 6 e 7 que a simulação do robô seguidor de linha após otimização PSO pode ser considerada um sucesso visto a trajetória traçada na figura 5.

Figura 4: Best Quality convergence da simulação do robô seguidor de linha após otimização PSO.

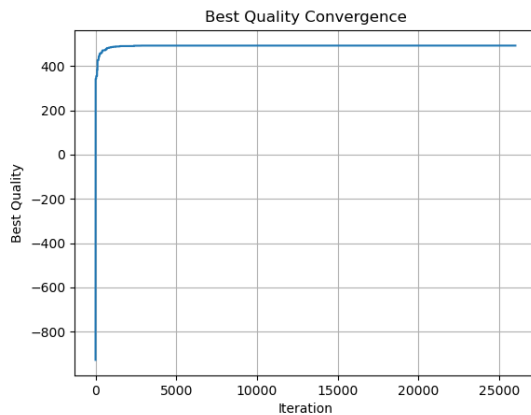


Figura 5: Trajetória do robô seguidor de linha após otimização com PSO.

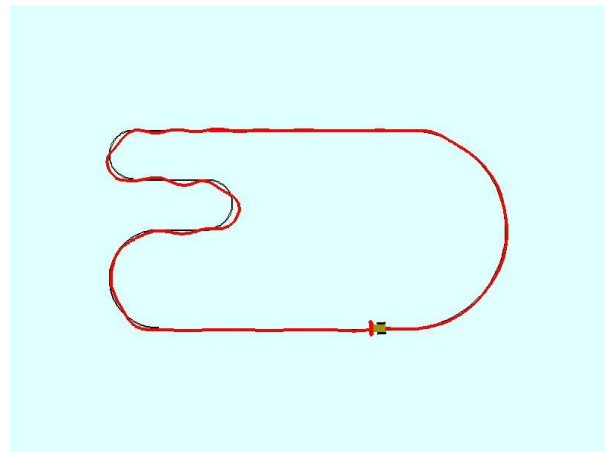


Figura 6: Convergência dos parâmetros da simulação do robô seguidor de linha após otimização PSO.

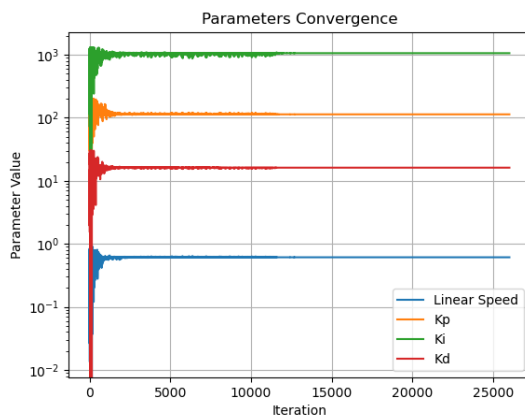


Figura 7: Quality convergence da simulação do robô seguidor de linha após otimização PSO.

