Instituto Tecnológico de Aeronáutica
CT-213: Inteligência Artificial para Robótica Móvel
Lab 04 - Otimização com Métodos Baseados em População

Bruno Benjamim Bertucci - Turma 23.2

1 Introdução

Com o intuito de evitar a tendência dos algorítmos de otimização de busca local de ficarem presos em mínimos locais, o que prejudica a eficácia deles na resolução de problemas de robótica, existem alternativas que resolvem esse problema ao considerar uma variedade de candidatos a solução ótima, denominada população.

Um dos algorítmos de otimização baseado em população é a Otimização do Enxame de Partículas. A eficácia desse algorítmo foi testada com a otimização de parâmetros de controle de um robô seguidor de linha.

2 Otimização do Enxame de Partículas

O algorítmo de Otimização do Enxame de Partículas, ou *Particle Smarm Optimization* (PSO), é um algorítmo que busca mínimos ou máximos globais através da distribuição de partículas, e do cálculo dos valores da função que se quer otimizar nas posições ocupadas por elas. Cada partícula armazena o melhor valor encontrado por ela, e o melhor valor encontrado entre todas as partículas também é armazenado.

As partículas mudam sua posição segundo uma velocidade dada a cada uma delas, sendo o cálculo dessa velocidade composto por fatores individuais de cada partícula, levando em conta a diferença entre a melhor posição encontrada por ela própria, e a posição atual dela, e também por fatores coletivos do enxame, considerando a diferença entre a melhor posição encontrada por ele e a posição atual de cada partícula. Com esse mecanismo, o algorítmo é capaz de encontrar máximos e mínimos globais.

2.1 Implementação

A Otimização do Enxame de Partículas foi implementada considerando a busca por maximização de uma função, e foi feita da seguinte forma: Primeiramente, é criado um determinado número de partículas. Inicialmente, a posição e a velocidade de cada uma são amostrados aleatóriamente segundo distribuição uniforme, dentro de limites mínimos (l) e máximos (u), de tal forma que, para a posição inicial x_i, temos que $x_i \in [l, u]$, e para a velocidade inicial, v_i, temos que $v_i \in [-(u-l), (u-l)]$. Cada partícula tem, também, duas variáveis para armazenar o valor máximo da função encontrado por essa partícula e a posição onde foi encontrado esse valor.

O algorítmo então itera cada uma das partículas, encontrando o valor da função para a posição atual de cada partícula. Em cada iteração feita no enxame inteiro, é armazenado o valor máximo encontrado por ele nessa iteração e a sua posição correspondente, e também é armazenado o valor máximo e a sua posição durante toda a execução do algorítmo, ou seja, o máximo global encontrado. Na iteração de cada partícula, é verificado se o valor atual da posição dela tem valor maior que o máximo encontrado na atual iteração do enxame, ou maior que o máximo global encontrado no algorítmo, substituindo eles caso essas condições sejam atendidas.

Quando todas as partículas do enxame tiverem sido analisadas, é feito o avanço do enxame. Cada partícula tem a sua velocidade atualizada segundo a Equação 1, e a posição atualizada em seguida de acordo com a Equação 2. v_i representa a velocidade atualizada, $v_{i,0}$ é a velocidade antes de ser atualizada, x_i é a posição atualizada, $x_{i,0}$ é a posição antes de ser atualizada, b_i é a melhor posição encontrada por essa partícula, b_g é a melhor posição global do algorítmo, ω é uma constante de inércia da velocidade, φ_p, φ_g são hiperparâmetros denominados parâmetros

cognitivo e social, e r_p, r_g são números no intervalo $[0, 1]$, amostrados aleatóriamente segundo distribuição normal a cada vez que o avanço do enxame é executado.

$$v_i = \omega v_{i,0} + \varphi_p r_p (b_i - x_{i,0}) + \varphi_g r_g (b_g - x_{i,0}) \tag{1}$$

$$x_i = x_{i,0} + v_i \tag{2}$$

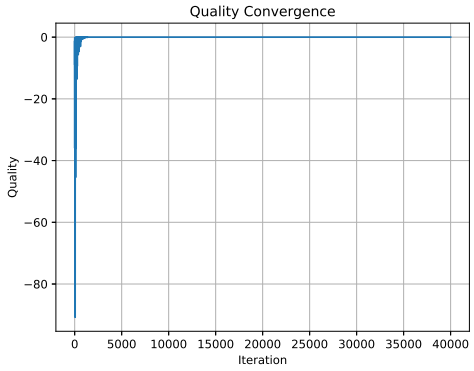
Após ser completado o avanço das partículas, é avaliado se a melhor posição encontrada nessa iteração pelo enxame é melhor que o máximo global encontrado até então, substituindo-o caso essa condição seja verdadeira. Em seguida, é iniciada uma nova iteração por todas as partículas do enxame, portanto as variáveis contendo o maior valor encontrado pela iteração e a sua posição são reiniciadas.

2.2 Teste do PSO

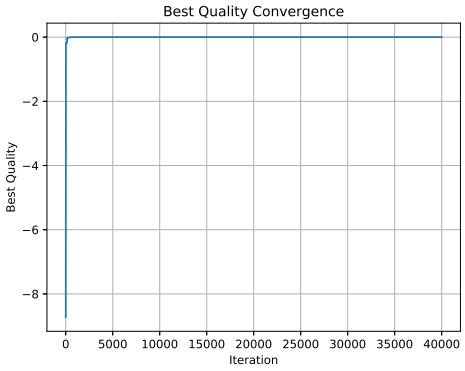
A implementação detalhada acima foi testada com uma função dada ela Equação 3.

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

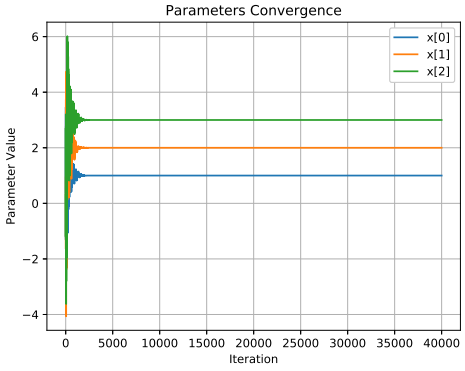
O máximo global dessa função pode ser determinado analiticamente, e é igual a $[1, 2, 3]^T$. Conforme esperado, o algorítmo PSO convergiu exatamente para essa solução, comprovando a sua funcionalidade. A Figura 1 ilustra o progresso da otimização feita pelo algorítmo.



(a) Convergência da qualidade da solução.



(b) Convergência da melhor qualidade da solução.



(c) Convergência dos valores dos parâmetros.

Figura 1: Evolução do teste do algorítmo PSO.

3 Implementação da avaliação do robô seguidor de linha

Após passar pelo teste realizado anteriormente, o algorítmo PSO implementado foi utilizado para otimizar um controlador PID (Proporcional, Integrativo e Derivativo) para um robô seguidor de linha. O controlador PID é utilizado para obter a velocidade angular que o robô deve ter em um determinado instante para corrigir a sua posição com relação à linha que ele deve seguir.

O valor da velocidade angular para um conrolador PID é dada pela Equação 4, onde e é o erro da posição do robô com relação à linha, dado pelo centro de massa das medidas do seu *array* de 7 sensores, segundo a Equação 5.

$$\omega = K_p e + K_i \int ede + K_d \dot{e} \tag{4}$$

$$e = \frac{\sum_i x_i I_i}{\sum_i I_i} \tag{5}$$

Os parâmetros K_p, K_i e K_d, junto com a velocidade linear comandada ao robô, v, são os valores que precisam ser otimizados nesse problema. Logo, na otimização por PSO, a posição será um vetor dado por $x = [v, K_p, K_i, K_d]^T$.

A função usada para medir a qualidade de cada candidato a solução é dada pela Equação 6, onde N é o número de iterações de um episódio de treinamento, v_k é a velocidade linear do robô, r_k é um vetor unitário que aponta na mesma direção que o robô no instante k, t_k é o vetor tangente à atual posição do caminho no instante k, w é um parâmetro, para o qual foi adotado o valor de 0,5, e e_k é o erro no instante k. Nos instantes em que o robô não detecta a linha, o valor de e_k foi considerado como uma constante igual a 3, significativamente maior do que o valor máximo detectável do erro, que é 0,03, a fim de penalizar mais as soluções que levam-no a sair completamente de baixo da linha.

$$f(x) = \sum_{k=1}^{N} (v_k(r_k \cdot t_k) - w \mid e_k \mid) \tag{6}$$

4 Resultados da otimização do controlador PID

O PSO foi executado para otimizar os parâmetros do controlador PID, e após 1.000 episódios de treinamento, sendo que, para cada episódio, uma solução diferente é testada, foram obtidos resultados satisfatórios que permitiram o robô seguir a linha com precisão suficiente. A trajetória seguida por ele usando a melhor solução encontrada no treinamento (Figura 2) mostra que ele é capaz de se manter relativamente próximo da linha, e consegue completar o percurso. Os parâmetros otimizados correspondem a $v = 0,6173$, $K_p = 136,4$, $K_i = 360,5$ e $K_d = 14,73$.

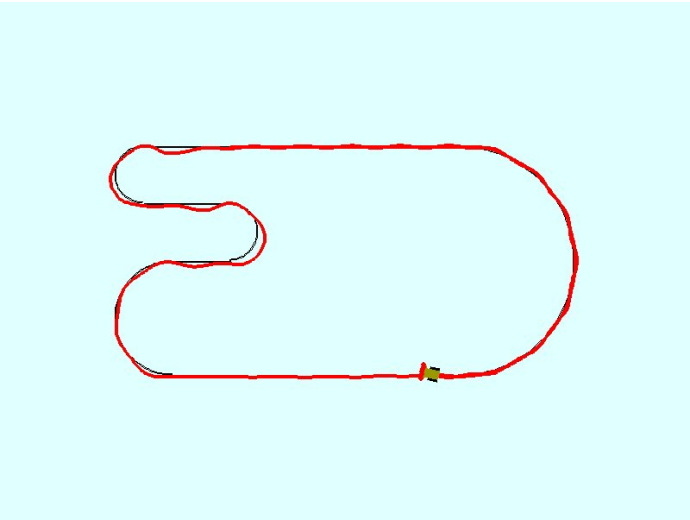
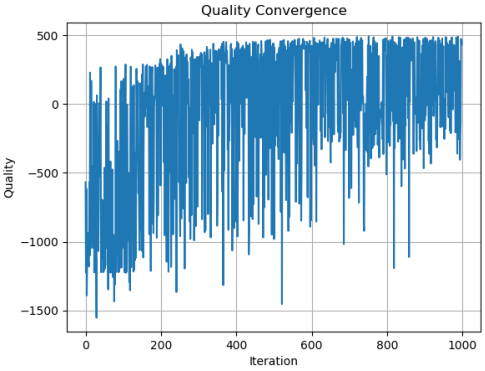


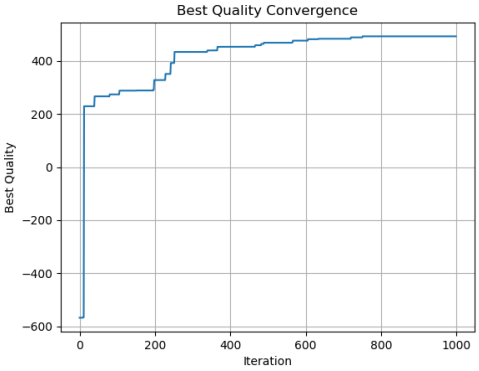
Figura 2: Trajetória otimizada do robô seguidor de linha.

Uma possível forma de reduzir ainda mais os desvios do robô com relação à linha é aumentar o valor do parâmetro w usado no cálculo da função de qualidade, a fim de penalizar mais esses desvios. Porém, é provável que a solução final do algorítmo tenha uma velocidade linear menor caso isso seja feito.

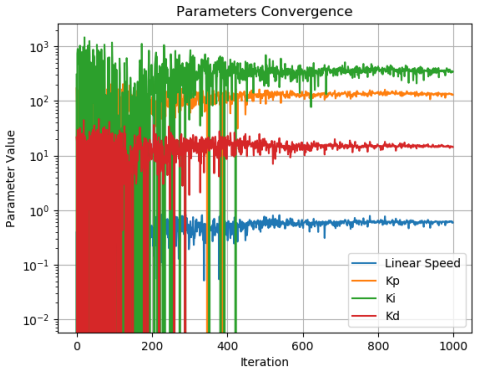
Finalmente, pode-se observar o progresso do processo de otimização na Figura 3. Observa-se a partir dele que mesmo após várias iterações, o algorítmo ainda encontra soluções com qualidade muito baixa. Ademais, a convergência da melhor solução encontrada, bem como dos valores dos parâmetros, começa a ocorrer após a execução de por volta de 500 iterações. Assim, conclui-se que o algorítmo de Otimização do Enxame de Partículas se mostrou adequado para o problema proposto de otimização de um controlador PID para um robô seguidor de linha.



(a) Convergência da qualidade da solução.



(b) Convergência da melhor qualidade da solução.



(c) Convergência dos valores dos parâmetros.

Figura 3: Evolução da otimização de controlador PID usando PSO.