

Planejamento de Rotas Baseado na Otimização por Nuvem de Partículas

Amanda Bragança Farias¹, Tiago Assunção Silva¹, Eduardo Max Amaro Amaral²

^{1,2}Coordenadoria de Informática – Instituto Federal do Espírito Santo (IFES)
Rodovia ES 010, Km 6.5, Manguinhos – 29.173-084 – Serra – ES – Brazil

¹Alunos do Curso Técnico em Informática - ²Orientador

amanda.estudo@hotmail.com, tiagoassun@gmail.com,
eduardomax@ifes.edu.br

1. Introdução

Robótica é a ciência de perceber e manipular o mundo físico através do computador controlando dispositivos mecânicos [Thrun et. Al 2000]. E nessa área uma das metas que se pretende alcançar é a de robôs autônomos, ou seja, robôs que são capazes de realizar tarefas as quais foram projetados sem a ajuda humana. O “path planning” (planejamento de caminho) é uma dessas tarefas e está relacionado a robôs móveis, onde o robô, diante de obstáculos, deve estar apto a escolher sozinho qual a melhor forma de seguir para alcançar qualquer ponto do ambiente.

A princípio, o planejamento de caminho para robôs pode aparentar ser algo simples, pois é um problema que está relacionado à percepção, algo que o ser humano, devido à inteligência elementar, possui e usa inconscientemente na sua interação com o ambiente que o cerca. Porém, as tarefas aparentemente simples tornam-se complexas de serem reproduzidas em robôs controlados por computador.

Neste contexto, dado o desafio de fazer um robô realizar a tarefa de planejar uma boa trajetória, a motivação deste trabalho é apresentar uma solução para isso, onde, dado o mapa do ambiente, será simulado o caminho que o robô planejará para seguir um percurso de custo mínimo até o destino. Para isso, foi feito um estudo dos métodos de navegação existentes e, após escolher uma estratégia, foi montado um algoritmo no Matlab (software destinado à análise e modelagem de sistemas e algoritmos, criado pela Mathworks em 1984) baseado na otimização por Nuvem de Partículas (Particle Swarm Optimization – PSO).

O PSO é uma técnica de computação evolucionária baseada em dinâmica de populações. Foi desenvolvida por James Kennedy, um psicólogo social, e por Russell Eberhart, um engenheiro elétrico, em 1995, inspirada na simulação de um sistema social simplificado. A intenção original era simular graficamente o comportamento de um bando de pássaros em voo com seu movimento localmente aleatório, mas globalmente determinado. Este método consiste na otimização de uma função objetivo através da troca de informações entre indivíduos (partículas) de uma população (enxame).

A Seção 2 deste trabalho descreve o algoritmo do PSO. A Seção 3 apresenta o algoritmo utilizado neste trabalho, mostrando alguns ajustes do algoritmo clássico do PSO e resultados. E a Seção 4 apresenta as conclusões.

2. Algoritmo PSO

O algoritmo de otimização por nuvem (ou enxame) de partículas (PSO) surgiu de experiências com algoritmos que modelam o “comportamento social”, observado em muitas espécies de animais, como, por exemplo, o movimento dos pássaros ou peixes.

Sendo assim, fazendo uma analogia desse algoritmo com o comportamento de bandos de pássaros, a busca por alimentos e a interação entre aves ao longo do voo são modeladas como um mecanismo de otimização. O termo **partícula** é adotado para simbolizar os pássaros e representar as possíveis soluções do problema a ser resolvido. A área sobrevoada pelos pássaros é equivalente ao **espaço de busca** e encontrar o local com comida, ou o ninho, corresponde a encontrar a **solução ótima**. Para que o bando de pássaros sempre se aproxime do objetivo, ao invés de se perder ou nunca alcançar o alvo focado, utiliza-se o indicador denominado **fitness**, função que irá avaliar o desempenho das partículas. Para alcançar o alvo focado, sejam os alimentos ou os ninhos, os pássaros fazem uso de suas experiências e da experiência do próprio bando. O termo indicador da experiência ou conhecimento individual de cada partícula, isto é, seu histórico de vida, é o **pbest**. Em uma abordagem mais simples, o responsável por representar o conhecimento do enxame como um todo é o **gbest**. A (Tabela 1) apresenta de forma resumida as nomenclaturas descritas acima:

Tabela 1. Identificação dos termos do PSO.

Termo	Significado
Partícula	Pássaro
Enxame	Bando de pássaros
Espaço de Busca	Área sobrevoada pelos pássaros
Posição	Localização de cada pássaro durante o voo
Solução ótima	Localização do pássaro onde ele encontrou o alimento ou o ninho
Fitness	Função de avaliação
Pbest	Melhor posição conhecida pelo pássaro (Experiência individual)
gbest	Melhor posição conhecida pelo enxame (Experiência coletiva)

No algoritmo PSO, os indivíduos da população são representados por pontos, denominados de partículas, que “voam” em um espaço de busca \mathbb{R}^d , onde d é a dimensão do espaço. As variações nos atributos dos pontos resultam na movimentação destes no espaço.

A maior parte dos algoritmos de PSO empregam dois princípios sócio-métricos que causam grandes influências no movimento das partículas: a influência social (g_{Best}), onde as partículas do enxame são diretamente influenciadas pelos “melhores”

indivíduos do grupo; a influência individual (p_{Best}), onde cada partícula é influenciada diretamente pelas suas experiências positivas.

Ambas as métricas são medidas por uma função de avaliação ($f(p)$), também chamada função objetivo ou de aptidão (*fitness*), que corresponde a melhor solução encontrada até então.

Uma partícula p_i irá se mover em uma determinada direção com uma velocidade nova ($v_i(t+1)$), conforme sua velocidade anterior ($v_i(t)$) e as distâncias entre a sua posição atual ($x_i(t)$) e: sua melhor posição (p_{Best}) e a melhor posição do grupo (g_{Best}). Sendo assim, a fórmula que manipula a velocidade de cada partícula é:

$$v_i(t+1) = w * v_i(t) + \varphi_1 * rand_1(.) * (p_{Best} - x_i(t)) + \varphi_2 * rand_2(.) * (g_{Best} - x_i(t)) \quad (1)$$

Onde: φ_1 e φ_2 são constantes limitadas a um intervalo finito, em que Kennedy denomina-os como sendo respectivamente os componentes “cognitivo” (coeficiente de individualidade) e “social” (coeficiente de sociabilidade). $rand_1(.)$ e $rand_2(.)$ são duas funções aleatória no intervalo [0,1]. w é o fator de inércia.

Uma vez que a velocidade da partícula é calculada, a posição da partícula i na próxima iteração é estabelecida como uma influência aditiva da posição antiga e da velocidade calculada, sendo expressa por (2):

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Agora, com as informações dadas até este ponto, o algoritmo básico de otimização por enxame de partículas pode ser descrito brevemente utilizando os seguintes passos: dada uma população inicial de partículas, atualiza-se o vetor posição a partir do vetor velocidade de cada partícula até que se atinja o critério de parada pré-definido.

O pseudocódigo do algoritmo, em sua forma original, é descrito no Algoritmo 1.

Algoritmo 1: Pseudocódigo do PSO.

1. Determine o número de partículas P da população.
2. Inicialize aleatoriamente a posição inicial (x) de cada partícula p de P .
3. Atribua uma velocidade inicial (v) igual para todas as partículas.
4. Para cada partícula p em P faça:
 - (a) Calcule sua aptidão $fp = f(p)$.
 - (b) Calcule e melhor posição da partícula p até o momento (p_{Best}).
5. Descubra a partícula com a melhor aptidão de toda a população (g_{Best}).
6. Para cada partícula p em P faça:

(a) Atualize a velocidade da partícula pela fórmula:

$$v_i(t+1) = w * v_i(t) + \varphi_1 * rand_1(.) * (p_B - x_i(t)) + \varphi_2 * rand_2(.) * (g_B - x_i(t))$$

(b) Atualize a posição da particular pela fórmula:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

7. Se condição de término não for alcançada, retorne ao passo 4.

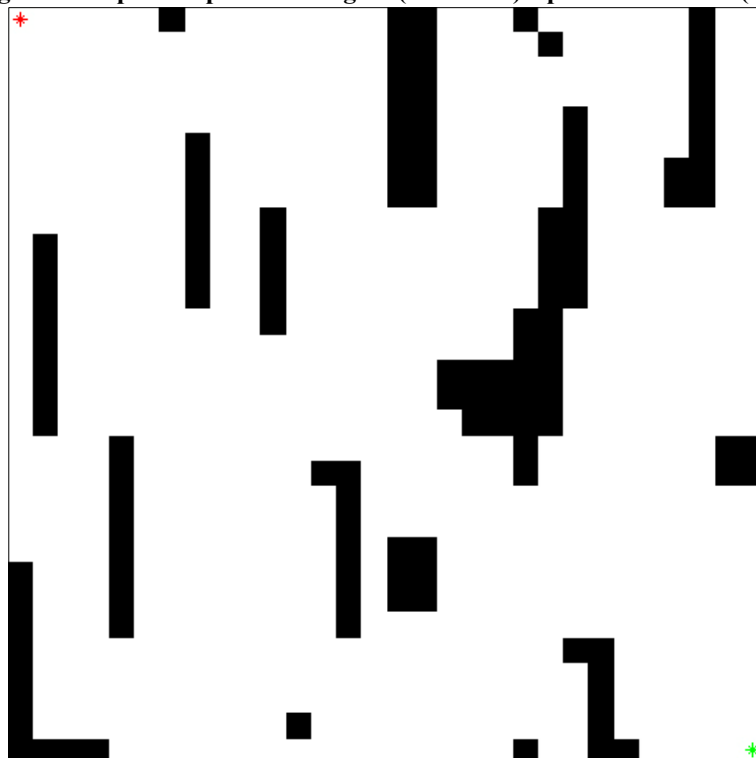
3. Aplicação ao problema do planejamento de rotas e resultados

Foram desenvolvidas e implementadas algumas estratégias neste trabalho com a finalidade de melhorar os resultados. Uma mudança da ideia clássica do PSO, por exemplo, foi que cada partícula não seria mais tratada como um ponto no espaço de busca e sim como o conjunto desses pontos gerados.

Então, dada uma população inicial de indivíduos, cada um com uma coordenada na matriz (mapa), somente a ligação entre eles que formaria uma partícula inteira e, sendo assim, formaria um caminho no mapa que seria avaliado como uma possível solução. A avaliação, neste caso, seria a busca da solução com o menor custo possível, o que torna a nossa função de fitness uma função de minimização.

Além disso, foi adotada a ideia de haver um parâmetro que tem a finalidade de dividir o mapa, desenvolvendo, assim, vários campos de busca, cada um com seus indivíduos. Para traçar o caminho completo, ou seja, formar a partícula, foi criado um ponto de interseção entre o primeiro campo, onde a coordenada do seu destino (que no momento é provisório) passa a ser a origem (também provisória) do próximo espaço de busca e assim segue. Sendo assim, é calculado e guardado em um vetor as melhores soluções de cada espaço de busca e, logo após, concatena no vetor as coordenadas já predefinidas: o ponto de origem real (de onde parte o robô) e o ponto de destino final do mapa.

Imagem 1. Mapa com ponto de origem (vermelho) e ponto de destino (verde)



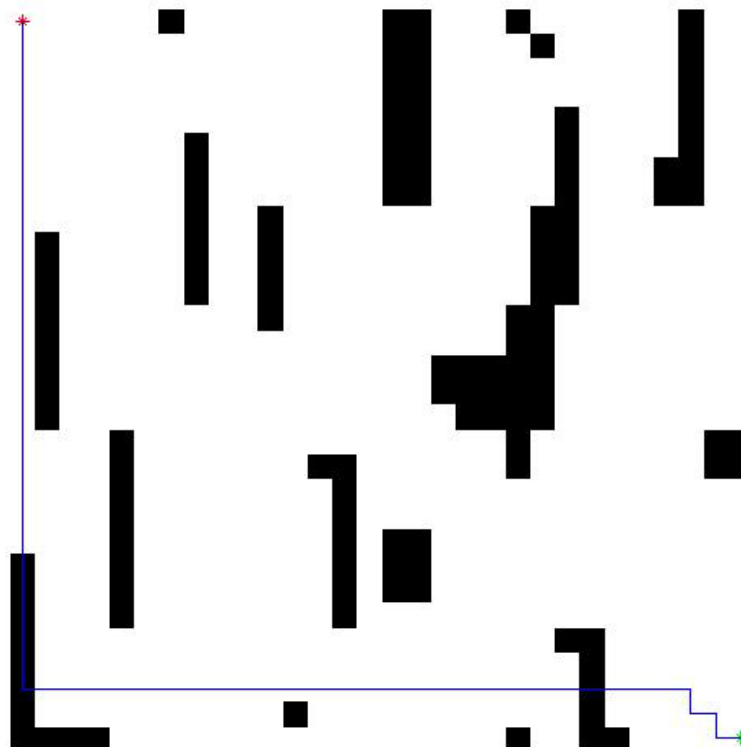
Verifique na (Tabela 2) alguns parâmetros de configuração utilizados para o algoritmo:

Tabela 2. Identificação dos parâmetros de configuração do algoritmo.

Parâmetro	Descrição
tam_enxame	Tamanho do enxame será a quantidade de partículas geradas.
max_iteracao	O número máximo de iteração que o loop principal do PSO irá repetir
tam_particula	Tamanho da partícula é a quantidade de pontos que cada partícula possui
parametroDivisao	É o valor que dividirá o mapa. Dado pela fórmula: $divisaoMapa = (Número de Colunas) / parametroDivisao);$

Foram realizados vários testes com os métodos explicados anteriormente e com diversos valores para os parâmetros que foram apresentados na (Tabela 2). Confira nas imagens a seguir, os resultados obtidos desses testes e na (Tabela 3) quais os valores que foram atribuídos para os parâmetros.

Imagem 2. Resultados - primeiro teste



A 2D environment with obstacles and a path. The environment is a white square with several black rectangular obstacles of varying sizes. A blue line represents a path starting from a red star at the top-left corner and ending at a green star at the bottom-right corner. The path is a step-like line that moves generally from top-left to bottom-right, avoiding the obstacles. A yellow shaded region highlights the area between the path and the obstacles, indicating a specific region of interest.

A 2D environment with obstacles and a path. The environment is a square with a white background. There are several black rectangular obstacles of various sizes and orientations. A blue line represents a path starting from a red star at the top left and ending at a green star at the bottom right. The path moves right, then down, then right again, navigating around the obstacles.

A 2D environment with obstacles (black rectangles) and a path (blue line) starting from a red star at the top left and ending at a green star at the bottom right. The path is a sequence of horizontal and vertical segments that navigate around the obstacles.

A 2D environment with obstacles and a path. The environment is a white square with several black rectangular obstacles of varying sizes. A blue line represents a path starting from a red star at the top-left corner and ending at a green star at the bottom-right corner. The path is a series of horizontal and vertical segments, navigating around the obstacles.

Imagem 7. Resultados - sexto teste

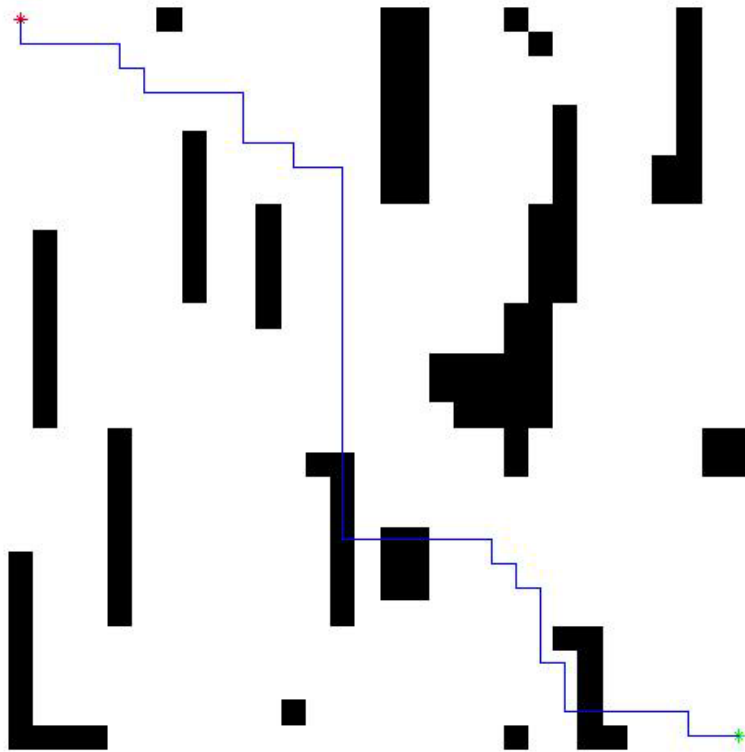


Tabela 3. Resultados – valores atribuídos aos parâmetros

tam_enxame	max_iteracao	tam_particula	parametroDivisao
100	200	2	1
100	200	2	4
70	200	2	6
100	200	2	6
100	100	4	4
100	200	5	2

Ao deparar com os resultados obtidos das simulações, observa-se que os planejamentos dos trajetos não foram exatamente como esperado, já que o robô não conseguiu desviar de todos os obstáculos. Porém, percebe-se que, para um projeto inicial e que não se encerra aqui, o algoritmo já está seguindo a lógica do que foi proposto para este projeto e já foi decidido que uma das possíveis soluções seria ampliar a parte do programa responsável pelas restrições para haver uma melhoria nas respostas dos planejamentos. Pretende-se, então, seguir com os testes e avaliações dos mesmos com a intenção de evoluir este projeto.

4. Conclusões e Trabalhos Futuros

Este trabalho apresentou um algoritmo baseado na otimização por Enxame de Partículas (Particle Swarm Optimization - PSO), com o objetivo de fazer com que um robô possa estar apto a encontrar por si só o melhor trajeto a seguir em um ambiente. Apesar dos resultados finais mostrarem que o problema não foi resolvido de vez, é possível perceber que a solução proposta se mostrou um bom caminho, podendo até usar o algoritmo feito como um guia de trajetória, principalmente se o robô for dotado de sensoriamento de desvio de obstáculos, o que a maioria é. Por isso, conclui-se que, como um projeto de inicial que terá continuidade, os resultados são promissores.

Além disso, o estudo desenvolvido para realizar este projeto proporcionou um melhor conhecimento sobre a área da robótica e da inteligência artificial, deixando, assim, uma grande possibilidade de exercitar os conhecimentos obtidos para evoluir nesse ramo e focar na busca de soluções para problemas encontrados até então, como também nas produções de trabalhos futuros que podem até proporcionar mais inteligência e autonomia a um robô.

5. Referências

Amaral, Eduardo M. A., Almeida, Álvaro Otto de, Simmer, Wesley Marcone e Trindade, Wagner Sartório da. Planejador de Rota baseado na Geometria de Manhattan. 2º Concurso de Trabalhos Técnicos, IV Computer on the Beach, Florianópolis, SC, 2013.

Thrun, S. and Fox, D. and Burgard, W. e Dellaert, F. Robust Monte Carlo Localization for Mobile Robots. *Jornal Artificial Intelligence*, vol. 128, nº 1-2, p. 99-141, 2000.

OTTONI, Guilherme de Lima e LAGES, Walter Fetter. Planejamento de trajetórias para robôs móveis, 2000.

Amaral, E. M. A. Uma Análise da Otimização por Enxame de Partículas e suas Aplicações.

Aloise, D.J., Oliveira, M.C.S., Silva, T.L., "Otimização Discreta por Nuvem de Partículas Aplicada ao Problema do Caixeiro Viajante", *GEPROS - Gestão da Produção, Operações e Sistemas*, Ano I, Edição 2, ISSN 1809-614X, abr/2006.