Otimização do algoritmo de path-planning: Univector Field Navigation

SSC0713 - Sistemas Evolutivos e Aplicados à Robótica

Leonardo Silva, Guilherme Caixeta, Anderson Hiroshi, e Lucas Luchiari

Robótica móvel

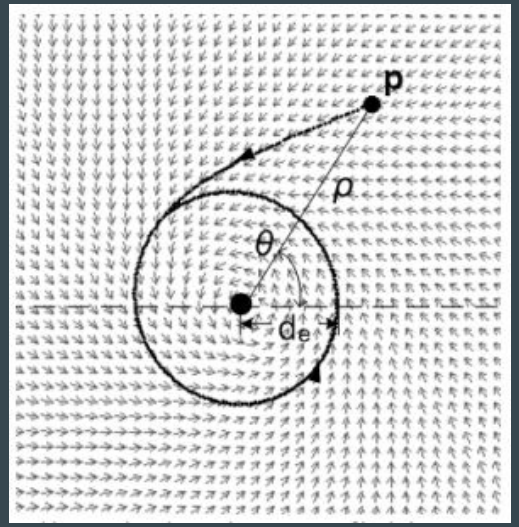
- Área com grande estudo na robótica
- Robôs com capacidade de mobilidade e autonomia
  - Oposição à robótica industrial (robôs manipuladores com base fixa)
- Nossa abordagem: **Futebol de robôs - RoboCup Small Size League (SSL)**
  - Ambiente extremamente dinâmico
  - Múltiplos robôs

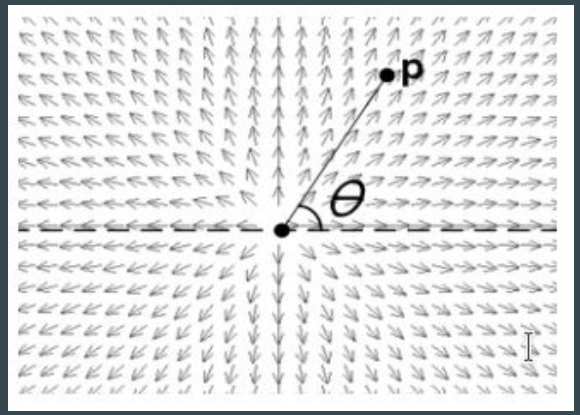
Univector Field Navigation

- Problema: dado um ponto de origem, uma orientação de origem, um destino, e uma orientação de destino desejada, qual caminho devo percorrer para atingir tal objetivo desviando dos obstáculos?
- Solução: algoritmo de path-planning para robôs móveis
  - Potential Fields
  - Oriented Potential Fields
  - **Univector field navigation**
  - Rapidly-exploring random tree (RRT)

Univector Field Navigation

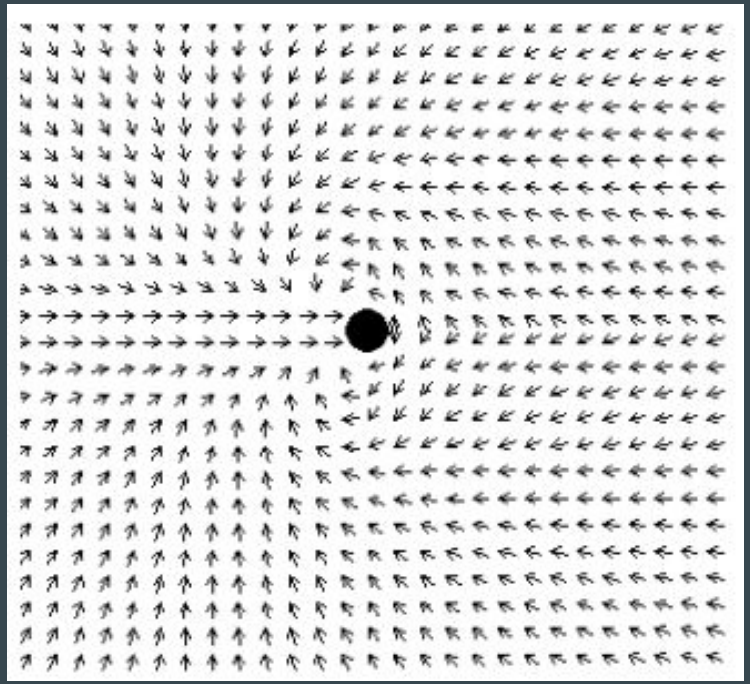
- Junção de 2 campos unitários principais
  - Move-to-goal univector field
  - Avoid-obstacle univector field





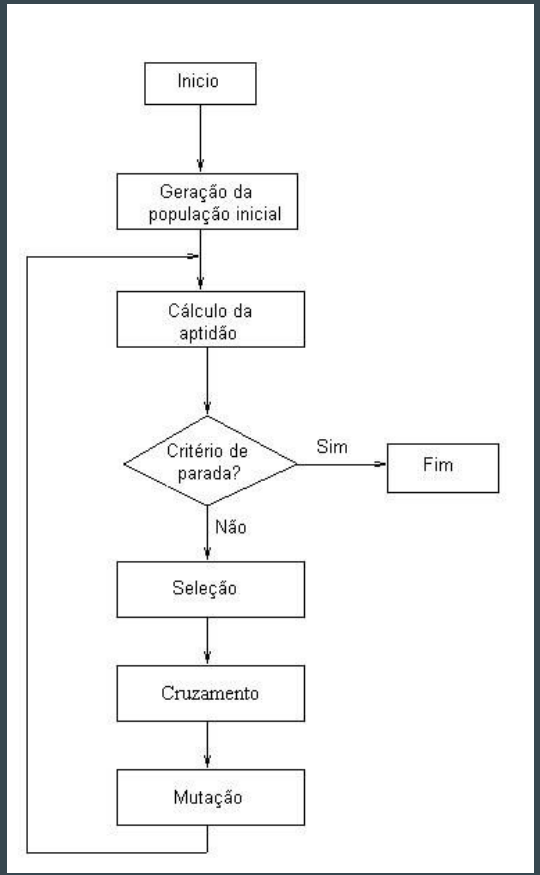
Univector Field Navigation

- Peculiaridade do UVF
  - Permite definir uma **orientação de chegada**
- Parâmetros
  - **de**: raio da espiral hiperbólica
  - **kr**: suavidade da espiral hiperbólica
  - **dmin**: raio de repulsão máxima
  - **delta**: ajuste da Gaussiana de repulsão
  - **maxLSpeed**: velocidade máxima linear
  - **maxASpeed**: velocidade máxima angular



Algoritmo Genético

A implementação básica de um algoritmo genético (AG) é como ilustrado na figura ao lado. Entretanto, foram necessário testes empíricos para **determinar qual tipo de crossover, mutação e seleção melhor solucionam o problema** em questão.



População Inicial

- 15 indivíduos (cromossomos)
- Para cada gene dos cromossomos é inicialmente gerado um número aleatório no intervalo [0, 5].

Seleção

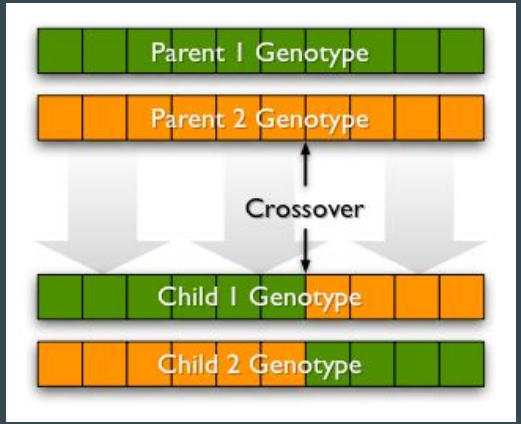
- A seleção é feita selecionando os melhores dentre o crossover e a mutação.
- Quando se selecionava da população pai, mesmo que de uma porcentagem dela, a solução convergia para um máximo local, não para o ótimo desejado.

Mutação

Para mutação, encontramos a melhor convergência utilizando um “passo” de +-[0.010, 0.075], sendo este aplicado aleatoriamente entre os genes com uma taxa de mutação de 60%.

CrossOver

O método de crossover é simples. Foi utilizado uma taxa de crossover de 50%.



Iterações

O código salva no disco, a cada iteração, o melhor indivíduo, o que permite parar o algoritmo a qualquer momento e exibir o resultado.

$$f1 = 100 reachedGoal \qquad f2 = \frac{1}{0.1 + linearError}$$

$$f3 = \frac{1}{0.1 + angularError} \qquad f4 = \frac{2}{0.1 + linearTime}$$

$$f5 = \frac{1.5}{0.1 + angularTime} \qquad f6 = \frac{1.5}{0.1 + entryAngError}$$

$$f7 = \frac{1.25}{0.1 + 10 Collisions}$$

$$fitness = f1 + f2 + f3 + f4 + f5 + f6 + f7$$

Fitness

Encontrar uma boa função de fitness foi a parte mais complexa do trabalho.

Pensando nos parâmetros do UVF e sabendo de antemão qual seria um resultado apropriado, determinamos a função de fitness conforme a imagem ao lado.

Resultados

Nº Indivíduos	Nº Iterações	Fitness	Tempo
15	10	147.92	6 minutos
15	20	146.56	11 minutos
50	10	147.95	13 minutos
50	20	147.35	31 minutos
15	**1600**	**148.17**	**7 horas**

Modo de Uso (parâmetros)

- É possível alterar diversos parâmetros na *main*:
  - population_size Tamanho da população
  - mutation_rate Taxa de mutação (entre 0 e 1)
  - crossover_rate Taxa de crossover (entre 0 e 1)
  - max_iterations Número de iterações

- Outros parâmetros do AG devem ser modificados diretamente no código.

Modo de Uso (execução)

Dependências:

- **Open Dynamics Engine (ODE)**: simulador do futebol de robôs do SSL.
- **Qt Library**: estruturas de dados em C++.

O projeto foi criado utilizando o QtCreator, é aconselhável utilizá-lo para execução.

Desenvolvimento

Durante o projeto utilizou-se o Github para implementação paralela entre os membros do grupo. O repositório se encontra em https://github.com/psilva-leo/UVF-GA onde é possível analisar todo o desenvolvimento.

Referências

- **Evolutionary Univector Field-based Navigation with Collision Avoidance for Mobile Robot**; Yusun Lim, Seung-Hwan Choi, Jong-Hwan Kim, Dong-Han Kim (http://www.sciencedirect.com/science/article/pii/S1474667016410293)
- **Fundamentos de Algoritmos Evolutivos**; Paulo Gabriel, Alexandre Delbem (http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_75.pdf)

Otimização do algoritmo de path-planning: Univector Field Navigation

SSC0713 - Sistemas Evolutivos e Aplicados à Robótica

Leonardo Silva, Guilherme Caixeta, Anderson Hiroshi, e Lucas Luchiari