

Enxames de Partículas

Particle Swarm Intelligence

Gisele L. Pappa

Introdução

- Enxames de partículas consistem em um conjunto de algoritmos inspirados no **comportamento natural de grupos**
- Criados para simular sociedades de indivíduos
 - Esses indivíduos estão trabalhando em um problema, e sendo influenciados por seus vizinhos

Inspiração

- Proposto por Eberhart and Kennedy em 1995, inspirado no comportamento social de pássaros e peixes



Introdução

- Inspiração vem de um conjunto de conceitos diferentes porém relacionados aos de insetos sociais:
 - Psicologia social
 - Interações entre indivíduos
 - Inteligência é resultado de interações entre os indivíduos (e adaptação)

Idéias Básicas - PSO

- Em PSO, partículas não tem autonomia ou especialização
- Paralelo com seres humanos
 - Interações levam a mudanças
 - Comportamento social ajuda os indivíduos de uma espécie a se adaptarem ao ambiente

Princípios básicos de adaptação cultural

(de acordo com a psicologia)

- **Avaliar** – todo organismo precisa avaliar estímulos
 - Pré-requisito para a aprendizagem – detectar estímulos positivos ou negativos
- **Comparar** – as pessoas se comparam umas com as outras
 - Essas comparações nos levam a melhorar
- **Imitar** – as pessoas tendem a imitar o comportamento de outras (especialmente das bem sucedidas)
 - Poucos animais são capazes de imitar uns aos outros

Idéia básicas - PSO

- Partícula representa um indivíduo com 2 tipos de informação:
 - **Experiência própria** (ele sabe que escolhas foram as melhores no passado)
 - **Conhecimento de outros indivíduos**, suas escolhas e sucesso associado
- Utiliza o conceito de uma partícula com uma velocidade se movendo num hiperespaço de busca (**velocidade e posição**)

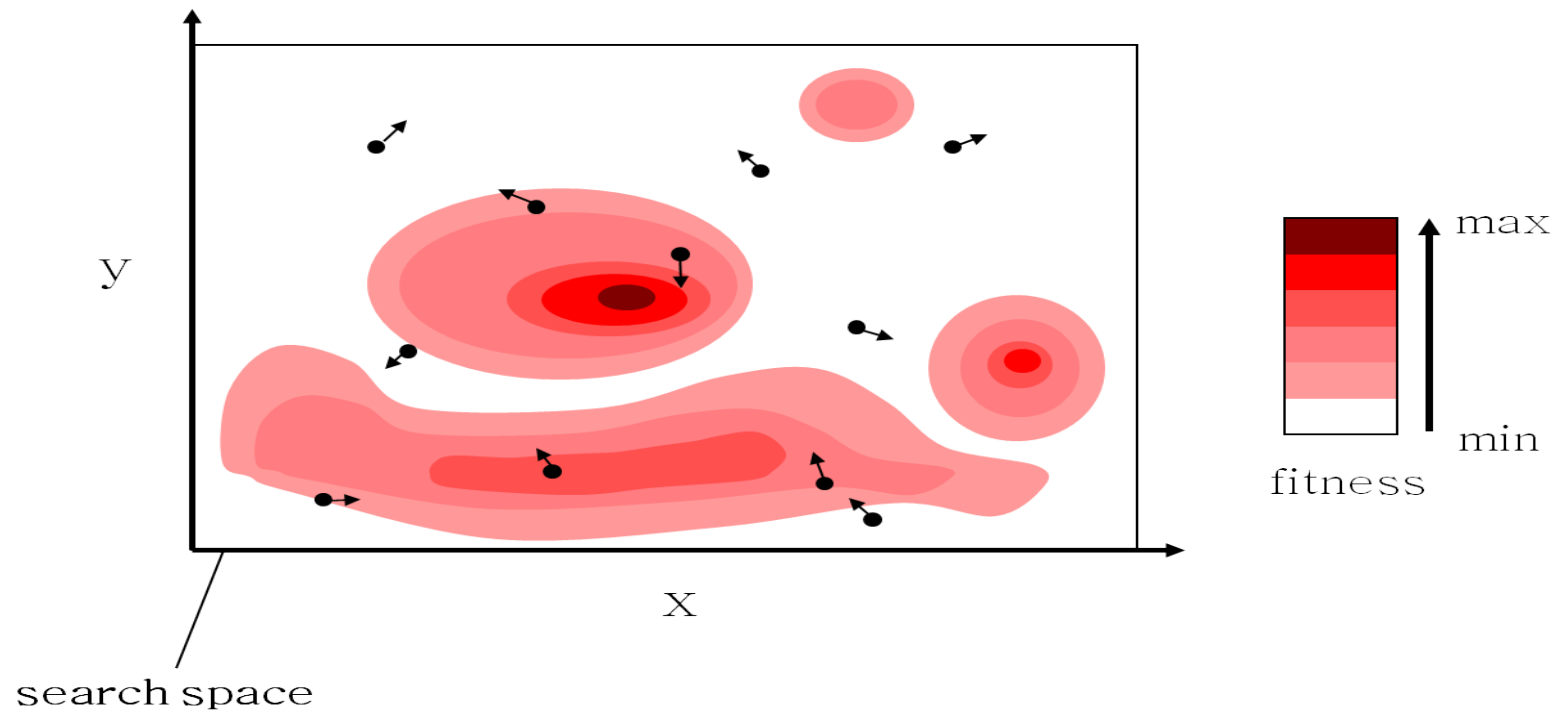
Componentes Principais

- Um conjunto de partículas
- Cada partícula representa uma solução candidata
 - Elementos da partícula representam parâmetros a ser otimizados
- Uma partícula possui
 - Uma coordenada no espaço de busca (\mathbf{x}_i)
 - Uma velocidade (v_i)

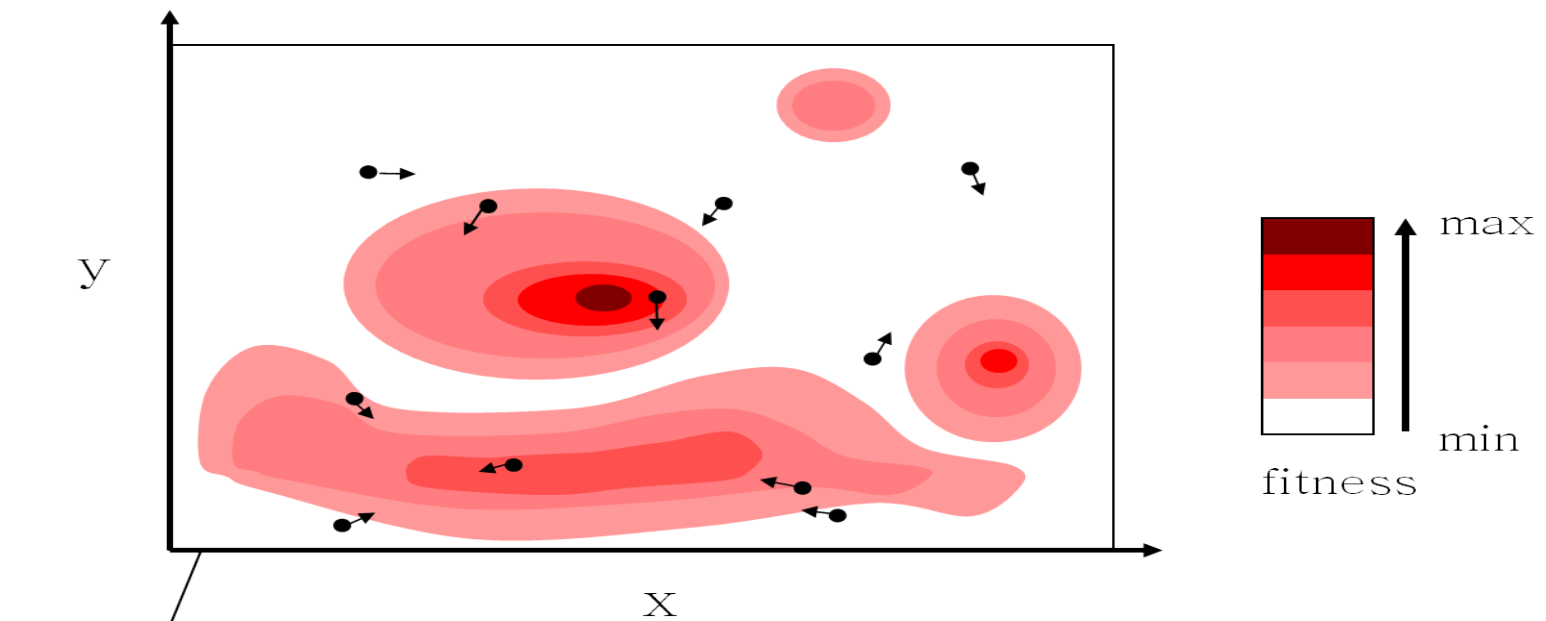
O algoritmo PSO

- **Velocidade** é responsável pelo processo de **otimização**
- **Esquecer ou aprender** são vistos como **aumentar ou diminuir o valor da posição da partícula**
- Conceito de melhor posição (fitness)
 - Altera a velocidade para nos levar a essa posição
- Indivíduos interagem com um número k de vizinhos

Exemplo

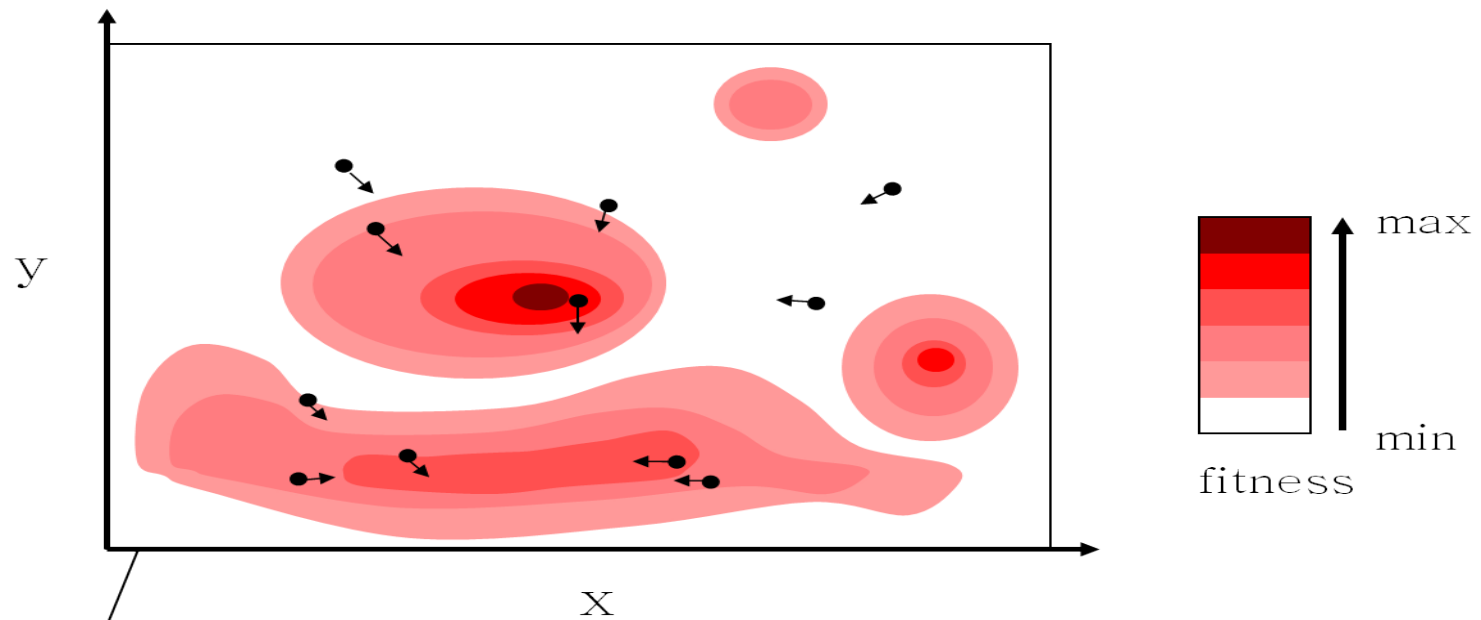


Exemplo



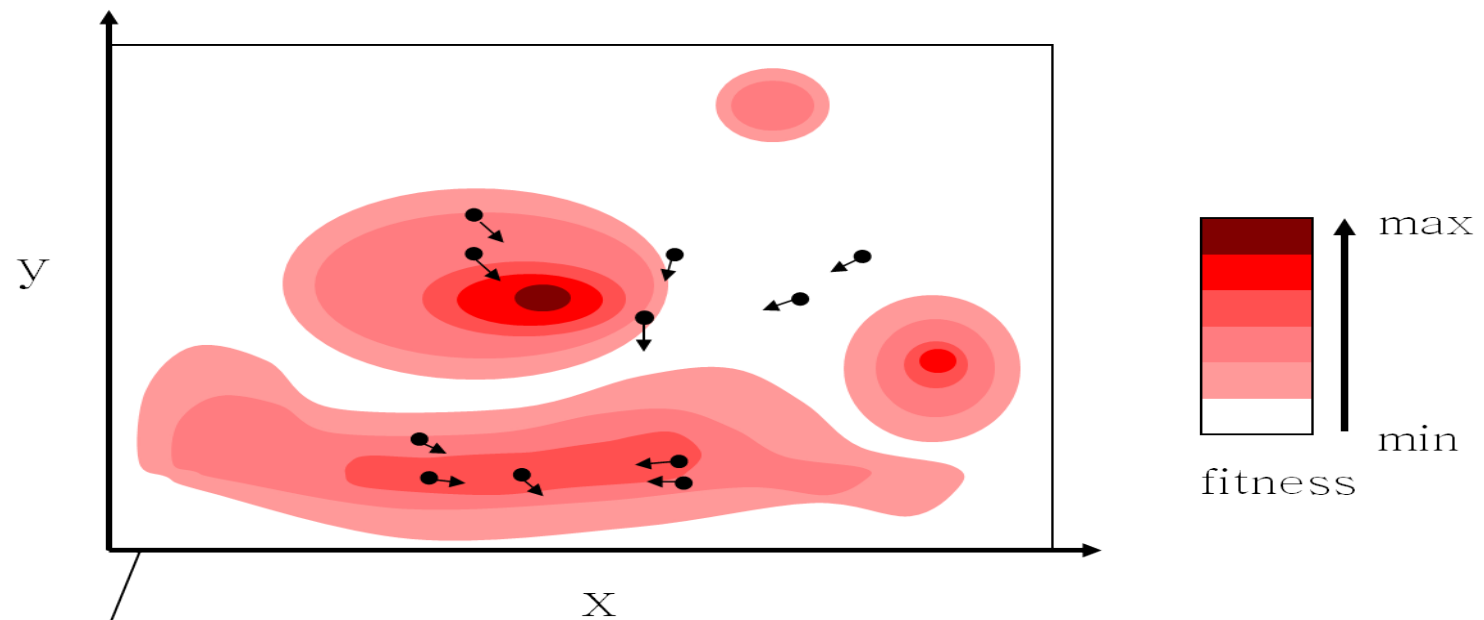
search space

Exemplo



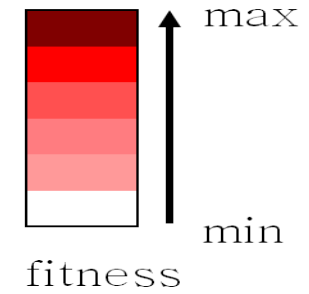
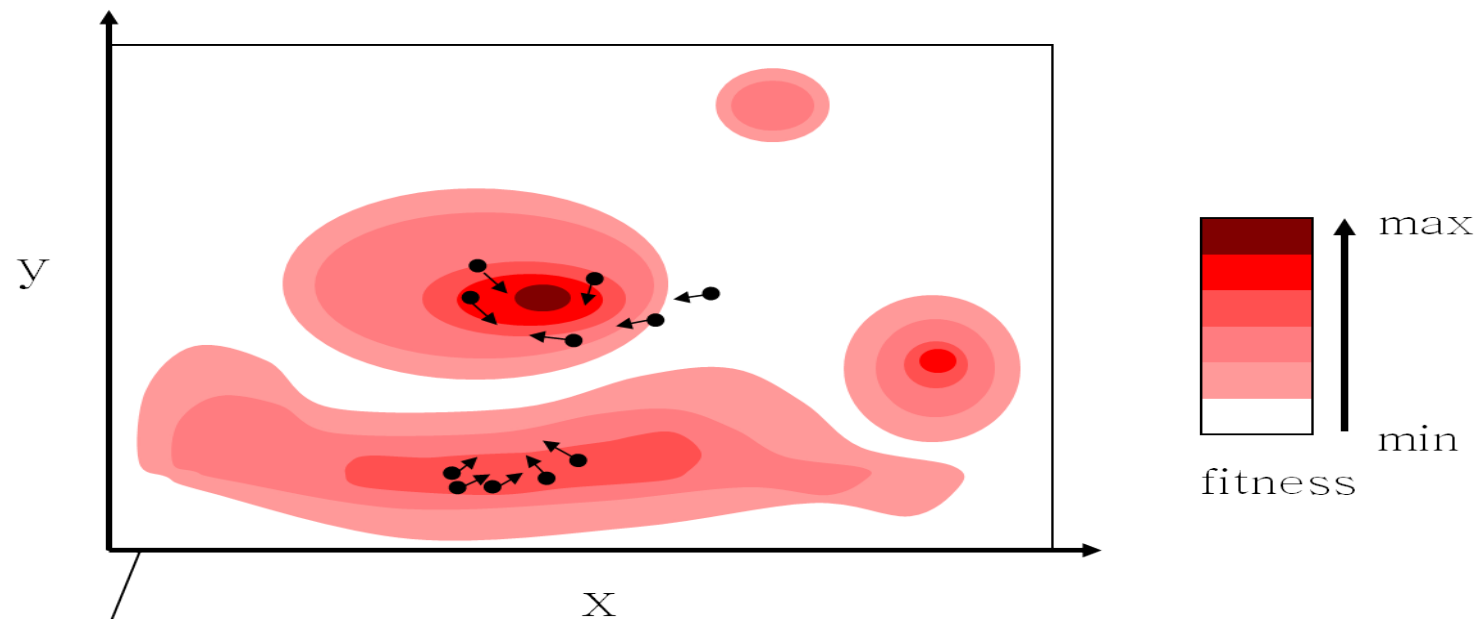
search space

Exemplo



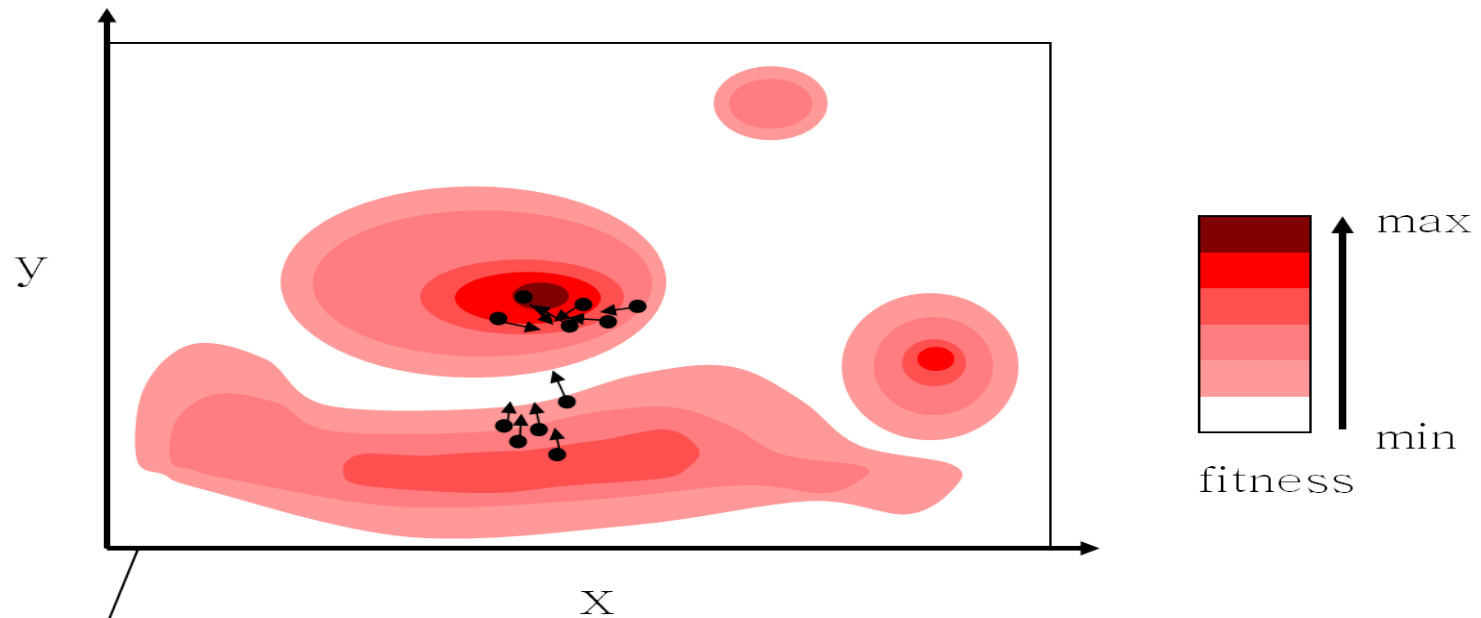
search space

Exemplo

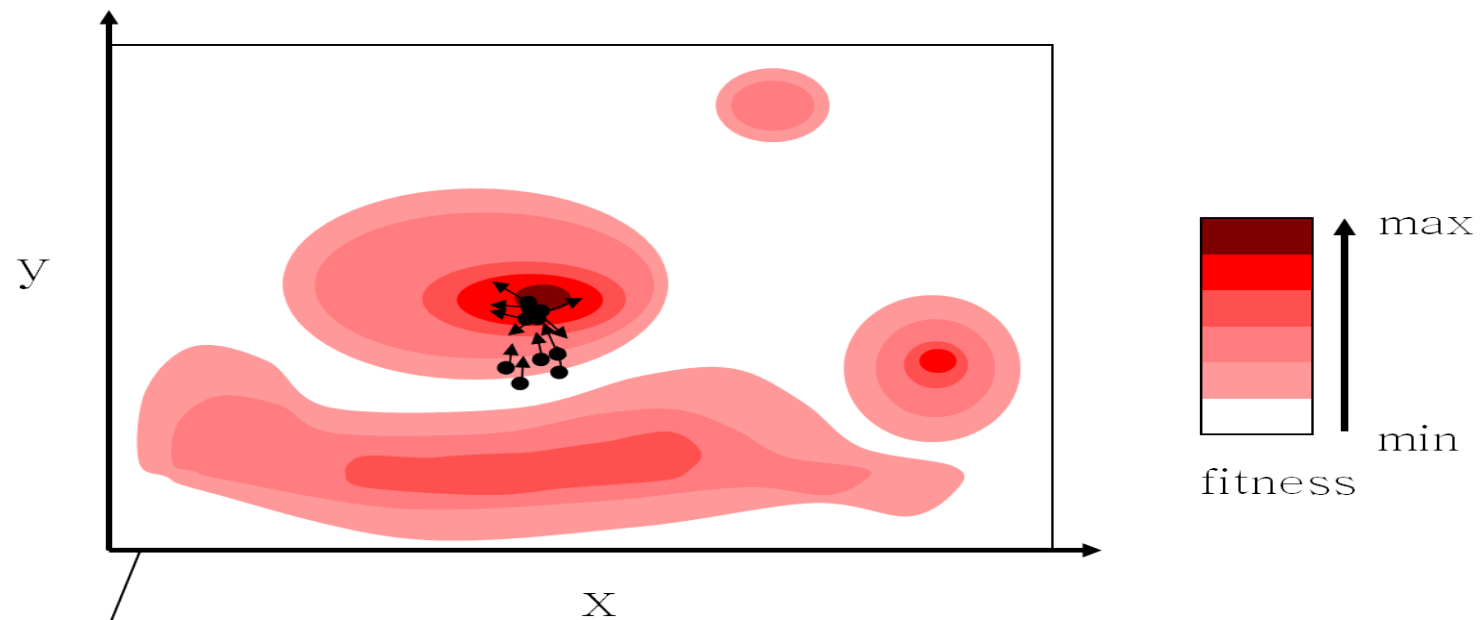


search space

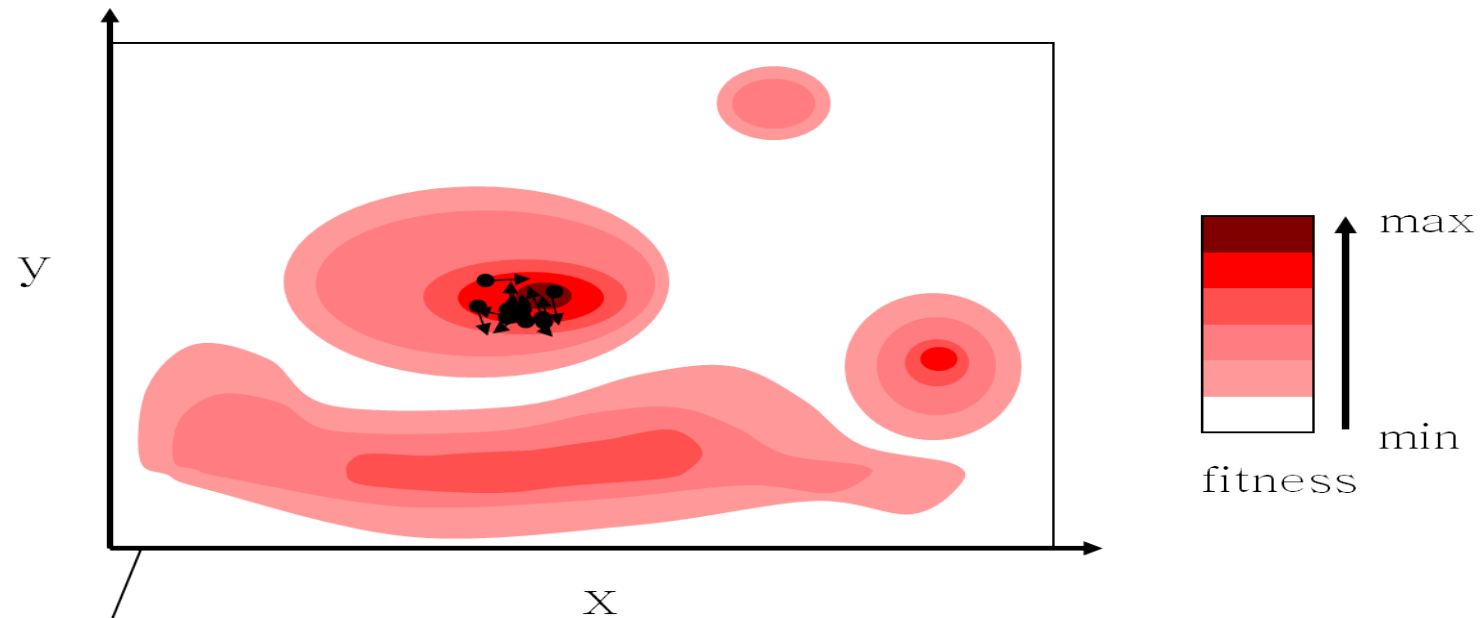
Exemplo



Exemplo

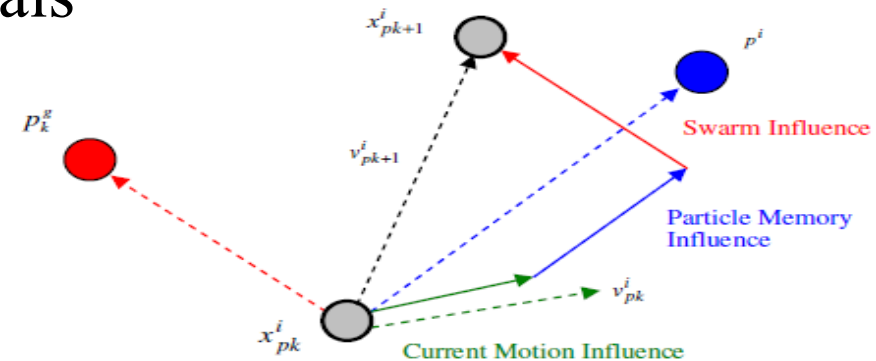


Exemplo

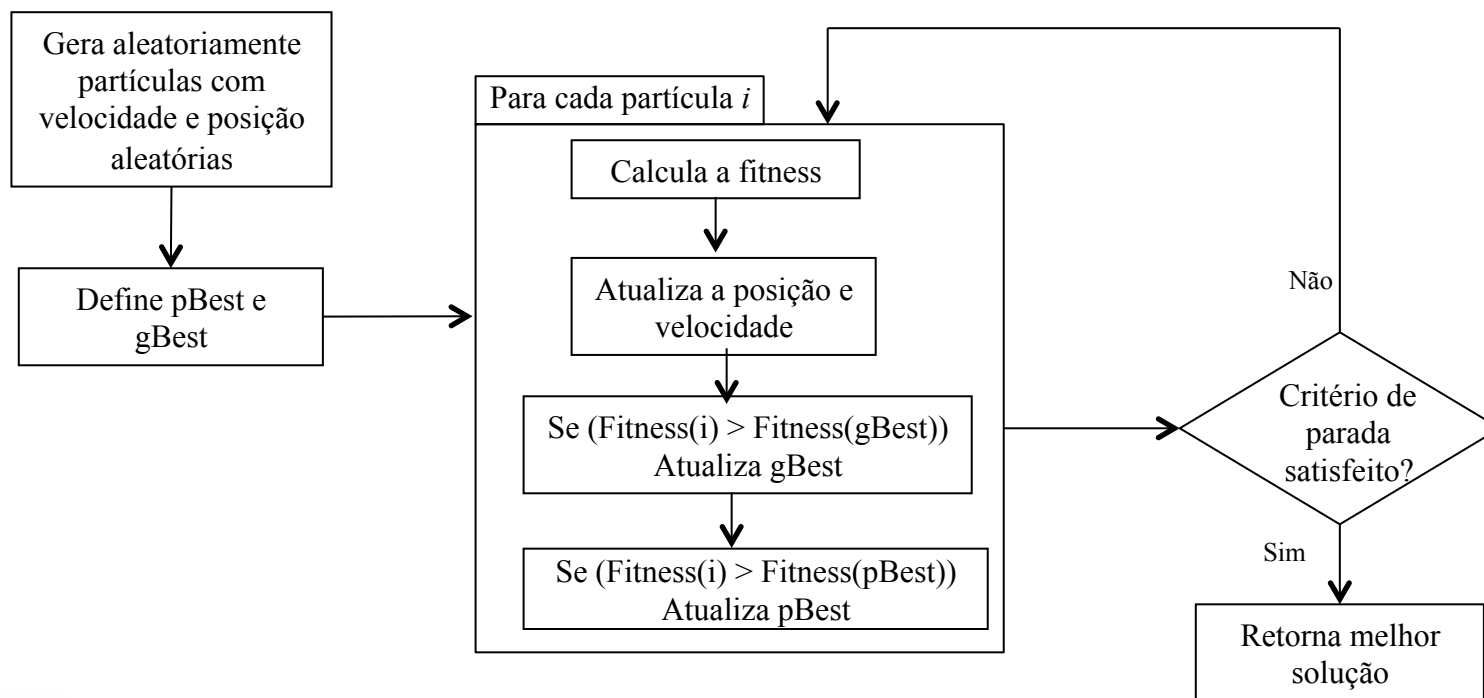


O algoritmo PSO

- Cada partícula ajusta sua velocidade dinamicamente de acordo com sua experiência e a dos vizinhos
- Cada partícula modifica sua posição de acordo com:
 - Posição e velocidade atuais
 - Distância para o *pbest*
 - Distância para o *gbest*



Fluxograma PSO



O algoritmo PSO

- Atualização da posição e velocidade da partícula:

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{v}_i(t)$$

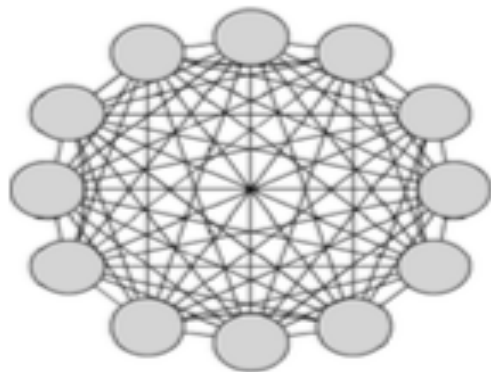
$$\mathbf{v}_i(t) = \omega \mathbf{v}_i(t-1) + c_1 \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t-1)) + c_2 \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t-1))$$

\mathbf{P}_i = local onde a melhor fitness do indivíduo foi encontrada (vetor cognitivo)

\mathbf{P}_g = local onde a melhor fitness do melhor vizinho foi encontrada (vetor social)

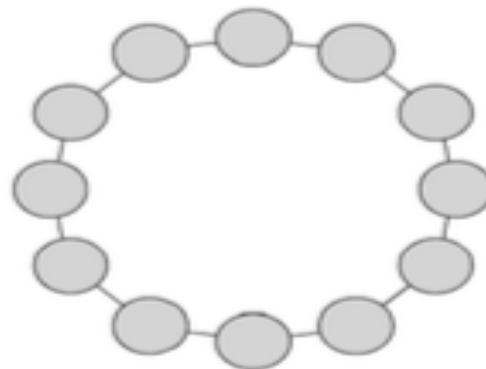
Vizinhança

- Diversos tipos:



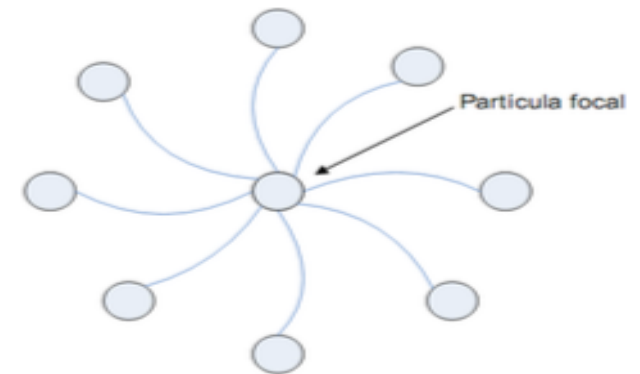
(a)

Estrela



(b)

Anel



Focal

A Velocidade – Modo 1

$$\mathbf{v}_i(t) = \omega \mathbf{v}_i(t-1) + c_1 \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t-1)) + c_2 \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t-1))$$

- i representa uma dimensão da partícula
- ω : constante de inércia (mantém partículas em movimento)
- $C_{1,2}$: constantes que afetam o quanto cada uma das melhores partículas afetam a partícula atual
- φ_1 escalar (taxa de aprendizagem)

A Velocidade – Modo 1

- ω : constante de inércia (mantém partículas em movimento)
 - Se ≥ 1 , velocidade aumenta ao longo do tempo
 - Swarm diverge
 - Partículas não conseguem mudar de posição para encontrar regiões mais promissoras
 - Se $0 < \omega < 1$, partículas desaceleram
 - Convergência depende dos valores de c_1 e c_2
- Normalmente é inicializado com 1
 - Linearmente reduzido ao longo das iterações

A Velocidade – Modo 1

- c_1 e c_2 : Controla *exploration* e *exploitation*
- *Exploration* – habilidade de explorar regiões do espaço de busca
- *Exploitation* – habilidade de se concentrar em uma região ao redor de uma área promissora e refinar a busca

A Velocidade – Modo 2

- Inércia é substituída por um fator de restrição χ (+
apropriado)

$$\mathbf{v}_i(t) = \chi \mathbf{v}_i(t-1) + c_1 \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t-1)) + c_2 \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t-1))$$

$$\chi = \frac{2}{|2 - \beta - \sqrt{\beta^2 - 4\beta}|}$$

$$\beta = c_1 + c_2$$

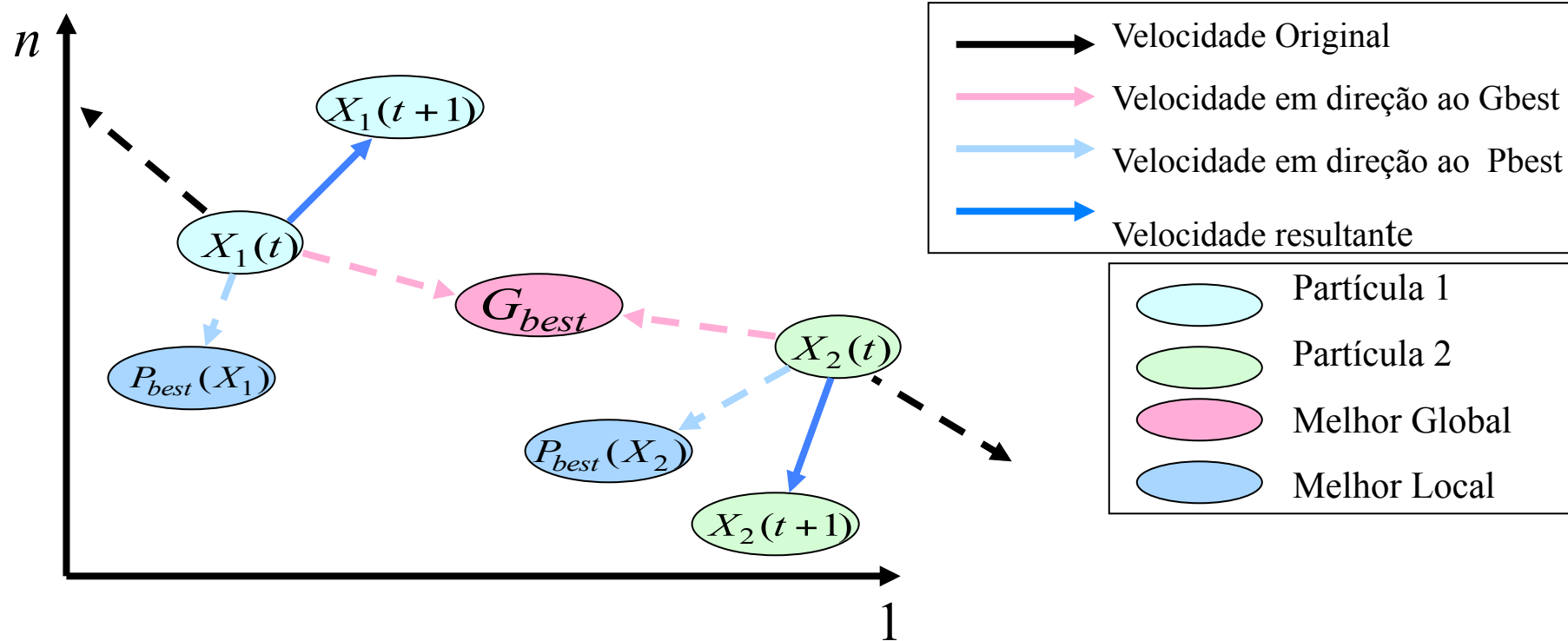
A Velocidade – Modo 2

$$\mathbf{v}_i(t) = \chi \mathbf{v}_i(t-1) + c_1 \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t-1)) + c_2 \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t-1))$$

$$\chi = \frac{2}{|2 - \beta - \sqrt{\beta^2 - 4\beta}|}$$

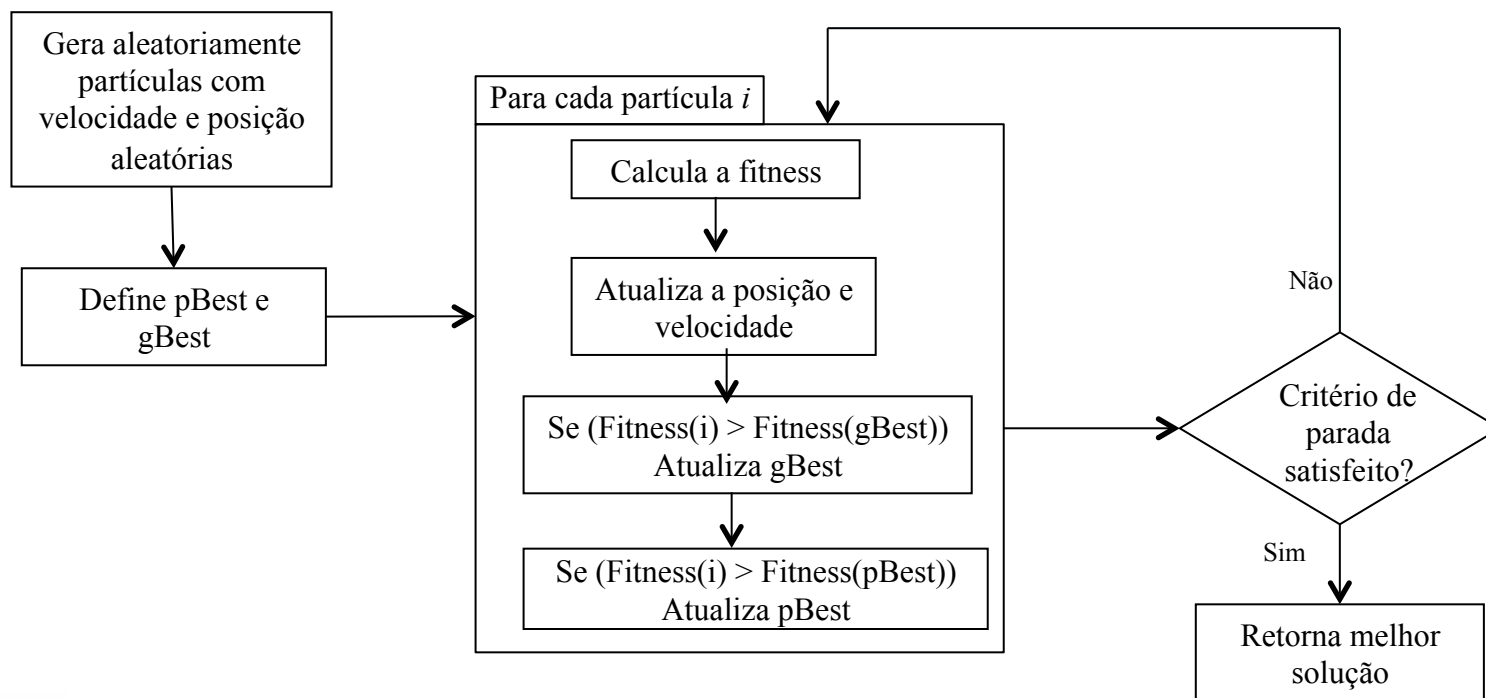
- Foi mostrado que:
 - $\beta < 4$ – enxame converge lentamente para a solução
 - $\beta > 4$ – convergência é rápida e garantida
- Assume-se c_1 e $c_2 = 2.05$ para garantir convergência

Movimentos das Partículas

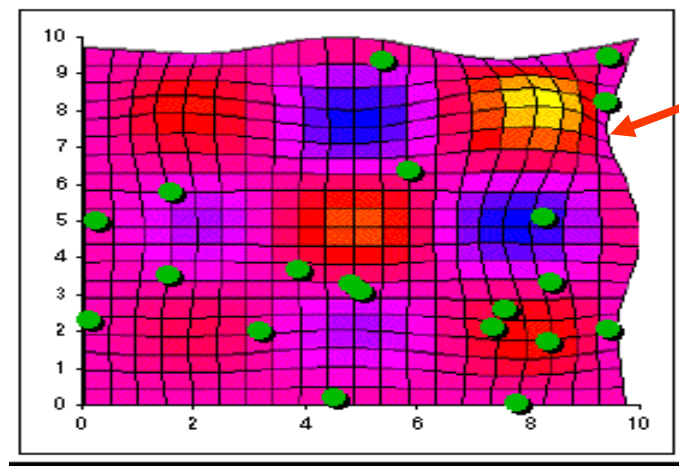


Partículas 1 e 2 vão em direção ao G_{best} e a seu P_{best} no espaço n -dimensional

Fluxograma PSO



Demo



Global
optimum

Tipos de PSO

- Existem dois tipos de algoritmos básicos
 - Um para dados contínuos
 - Mais utilizado e, conseqüentemente, mais estudado
 - Outro para dados binários

PSO para dados contínuos

- Uma **partícula** consiste em um vetor de números reais (valores das variáveis)
- Notação para a partícula i : $x_i = \langle x_{i1}, \dots, x_{ij}, \dots, x_{iD} \rangle$
onde x_{ij} = o valor da coordenada j da partícula i
e $j = 1, \dots$, número de variáveis (número de dimensões: D)

PSO para dados contínuos

- A **velocidade de uma partícula** é um vetor de números reais adicionado a posição da partícula em um espaço de tempo (uma iteração)
- Atualização da posição e velocidades da partícula:

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{v}_i(t)$$

$$\mathbf{v}_i(t) = \omega \mathbf{v}_i(t-1) + c_1 \varphi_1(\mathbf{p}_i - \mathbf{x}_i(t-1)) + c_2 \varphi_2(\mathbf{p}_g - \mathbf{x}_i(t-1))$$

PSO para dados contínuos

- Para evitar que a partícula saia do espaço de busca, devemos limitar a velocidade
- Para cada dimensão (variável) $j, j = 1, \dots, D$:

se $v_{ij}(t) > V_{\max}$ então

$$v_{ij}(t) = V_{\max}$$

senão se $v_{ij}(t) < -V_{\max}$ então

$$v_{ij}(t) = -V_{\max}$$

- A inércia minimiza esse problema

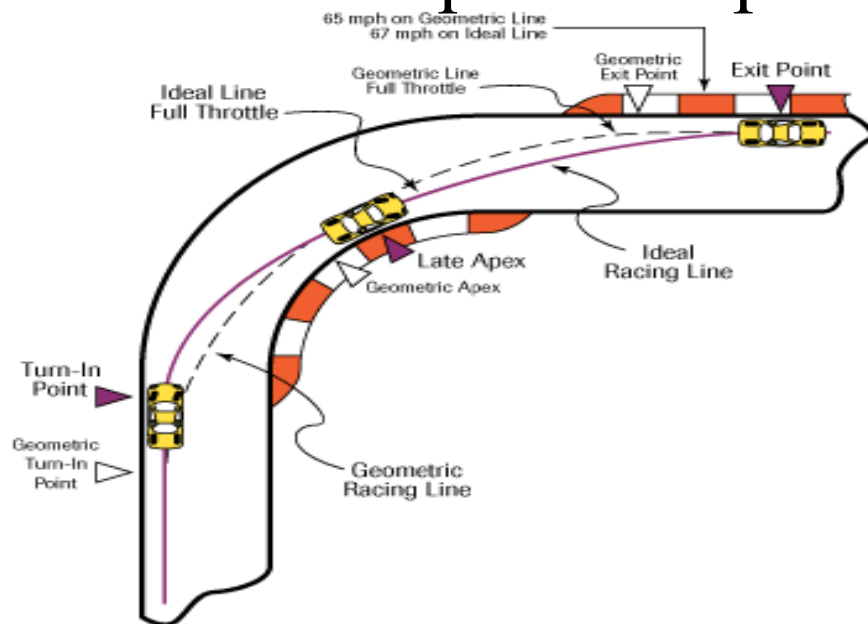
Parâmetros do PSO

- Número de iterações
- Número de partículas
 - (10—50) são normalmente suficientes
- C_1 (importância do vetor cognitivo)
- C_2 (importância do vetor social)
 - Usualmente iniciam seus valores com 2.05
- φ_1 e φ_2 - taxas de aprendizagem

Otimizando os parâmetros de um Jogo de Corrida

Objetivo

- Encontrar o melhor caminho a ser seguido pelos carros controlados pelo computador em um jogo de corrida



Parâmetros

- Cada partícula representa um conjunto de 10 parâmetros que representam o caminho a ser percorrido
- 20 partículas são distribuídas no espaço
- O algoritmo roda 100 iterações
- A fitness é dada pelo tempo que o carro leva para completar uma volta

Resultados

- Jogador humano leva em média 70s para completar uma volta
- Primeira iteração do algoritmo: melhor partícula tinha um tempo de 80s, e a média de todas as partículas era de 115 s
- Final da execução: a melhor partícula completava a volta em 63s, tornando o computador imbatível por um usuário humano

Bibliografia

1. <http://www.swarmintelligence.org/>
2. <http://www.swarmintelligence.org/tutorials.php>
3. Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995
4. http://www.gamasutra.com/features/20051213/villiers_01.shtml