

Algoritmos de Inteligência de Partículas

Rui Mendes

Universidade do Minho

Computação Natural

Sumário I

1 Introdução

- Algoritmo de Optimização
- Procura Local
- Procura Global

2 Origens

- Simulações de Impacto Social de Latané
- Modelo de Cultura de Axelrod
- Boids
- Boids que Procuram Comida

3 O Algoritmo

- Particle Swarm Inicial
- Esquema Geral
- O PSO inicial
- Geração de Soluções no PSO
- Velocidade Máxima
- Coeficiente Inercial

Sumário II

- Coeficiente de Constrição
- Equivalência entre os Modelos Inercial e de Constrição
- PSO Canónico
- Exemplo de Execução

4 Conceitos Importantes de PSO

- Conceitos Importantes de PSO
- Algoritmos Genéticos
- Procura Harmónica
- Evolução Diferencial
- Comparação com Evolução Diferencial
- Utilização do PSO
- Exemplos

5 Convergência Prematura

- Convergência Prematura
- Quais são as causas?
- Como resolver problemas de convergência prematura em PSO?

Sumário III

- Porque há convergência prematura em PSO?

6 Vizinhaças

- Porquê Utilizar Vizinhaças?
- Exemplo de Influência Imediata
- Conceito de Vizinhaça
- Exemplo do Fluxo de Influência
- Vizinhaça Cartesiana
- Vizinhaça Social
- Estatísticas de Grafos Importantes
- Características de uma Boa Topologia
- Exemplos de Topologias Razoáveis
- Topologias Dinâmicas

7 Imitação da Norma Social

- Imitação
- Normas Sociais
- Fully Informed Particle Swarm

Sumário IV

- Geração de Soluções no FIPS
- Pseudo-Algoritmo de Geração de Soluções do FIPS
- Diferenças no FIPS

8 O Dogma Central de PSO

- Bare-Bones PSO
- Gaussian FIPS
- FIPS Gaussiano com Baricentro Estocástico

9 Optimização Dinâmica

- Problemas de Optimização Dinâmica
- Vantagens do PSO
- Conceitos Base
- Implementação dos Conceitos

10 Fim

Algoritmo de Optimizaç o

Problema de Optimizaç o Real

Espaço de soluç es $x \in \mathbb{R}^n$;

Restriç es $g_i(x) \in \mathbb{B}$;

Funç o objectivo $f(x) \in \mathbb{R}$.

Caracter sticas de um Algoritmo de Optimizaç o

Exploraç o Procurar novas regi es independentemente da sua qualidade;

Aproveitamento Procurar unicamente em regi es de alta qualidade;

Elitismo S  aceitar soluç es melhores do que as existentes;

Passo Tamanho do passo usado em perturbaç es.

Características

- Procura melhores soluções na vizinhança da solução actual;
- Só usa aproveitamento;
- Normalmente usa um único agente.

Características

- Tem que balancear exploração e aproveitamento;
- Se usar exploração a mais não consegue soluções de boa qualidade;
- Se usar aproveitamento a mais arrisca-se a ficar preso em ótimos locais;
- Pode usar mais do que um agente.

Simulações de Impacto Social de Latané

Parâmetros

Cada agente muda a sua opinião de acordo com vários parâmetros:

- 1 Carisma;
- 2 Proximidade;
- 3 Número de fontes.

Resultados

Consolidação A diversidade das opiniões diminui pela exposição aos argumentos da maioria;

Segmentação Os agentes tendem a tornar-se mais semelhantes aos seus vizinhos;

Correlação As atitudes que eram inicialmente diferentes tornam-se parecidas;

Diversidade Devido à segmentação existem opiniões minoritárias.

Modelo de Cultura de Axelrod

Parâmetros

- Cada agente é uma string de símbolos;
- A probabilidade de interacção entre dois agentes depende da similaridade.

Resultados

Consolidação Tal como no modelo de Latané os agentes sofrem um processo de homogeneização;

Segmentação Como a interacção é entre agentes semelhantes formam-se várias classes ao longo da simulação;

Correlação As atitudes que eram inicialmente diferentes tornam-se parecidas;

Diversidade Não converge necessariamente para uma única string.

Boids

Regras

- Separação** Cada agente afasta-se dos outros se estiver próximo demais;
- Alinhamento** Cada agente tende para a média das direcções dos seus vizinhos;
- Coesão** Cada agente tende a deslocar-se para o centro de gravidade dos seus vizinhos.

Boids que Procuram Comida

Simulação efectuada por Kennedy e Eberhart.

Regras

- Existe um “ninho”;
- Cada agente é atraído pelo ninho;
- Cada agente memoriza a posição em que esteve mais próximo do ninho;
- Cada agente partilha com os seus vizinhos a informação sobre a posição mais próxima do ninho onde esteve.

Particle Swarm Inicial

A Ideia

- E se em vez da distância ao ninho se utilizasse uma função genérica?
- Poderia o enxame encontrar o ponto que tivesse a distância mínima?
- Seria possível utilizar esta ideia como um algoritmo de optimização?
- Esta ideia foi publicada por Kennedy e Eberhart em 1995.

Particle Swarm Inicial

A Ideia

- E se em vez da distância ao ninho se utilizasse uma função genérica?
- Poderia o enxame encontrar o ponto que tivesse a distância mínima?
- Seria possível utilizar esta ideia como um algoritmo de optimização?
- Esta ideia foi publicada por Kennedy e Eberhart em 1995.

Particle Swarm Inicial

A Ideia

- E se em vez da distância ao ninho se utilizasse uma função genérica?
- Poderia o enxame encontrar o ponto que tivesse a distância mínima?
- Seria possível utilizar esta ideia como um algoritmo de optimização?
- Esta ideia foi publicada por Kennedy e Eberhart em 1995.

Particle Swarm Inicial

A Ideia

- E se em vez da distância ao ninho se utilizasse uma função genérica?
- Poderia o enxame encontrar o ponto que tivesse a distância mínima?
- Seria possível utilizar esta ideia como um algoritmo de optimização?
- Esta ideia foi publicada por Kennedy e Eberhart em 1995.

Esquema Geral

- 1 Inicializar a população;
- 2 Avaliar a qualidade dos agentes;
- 3 Para cada agente:
 - 1 Escolher agentes na sua vizinhança;
 - 2 Imitar esses agentes;
 - 3 Actualizar a melhor experiência caso o valor actual seja melhor;
- 4 Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- 1 Inicializar a população;
- 2 Avaliar a qualidade dos agentes;
- 3 Para cada agente:
 - 1 Escolher agentes na sua vizinhança;
 - 2 Imitar esses agentes;
 - 3 Actualizar a melhor experiência caso o valor actual seja melhor;
- 4 Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- 1 Inicializar a população;
- 2 Avaliar a qualidade dos agentes;
- 3 Para cada agente:
 - 1 Escolher agentes na sua vizinhança;
 - 2 Imitar esses agentes;
 - 3 Actualizar a melhor experiência caso o valor actual seja melhor;
- 4 Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- ➊ Inicializar a população;
- ➋ Avaliar a qualidade dos agentes;
- ➌ Para cada agente:
 - ➊ Escolher agentes na sua vizinhança;
 - ➋ Imitar esses agentes;
 - ➌ Actualizar a melhor experiência caso o valor actual seja melhor;
- ➍ Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- ➊ Inicializar a população;
- ➋ Avaliar a qualidade dos agentes;
- ➌ Para cada agente:
 - ➊ Escolher agentes na sua vizinhança;
 - ➋ Imitar esses agentes;
 - ➌ Actualizar a melhor experiência caso o valor actual seja melhor;
- ➍ Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- ① Inicializar a população;
- ② Avaliar a qualidade dos agentes;
- ③ Para cada agente:
 - ① Escolher agentes na sua vizinhança;
 - ② Imitar esses agentes;
 - ③ Actualizar a melhor experiência caso o valor actual seja melhor;
- ④ Iterar para 2 se ainda não atingimos o critério de paragem.

Esquema Geral

- ① Inicializar a população;
- ② Avaliar a qualidade dos agentes;
- ③ Para cada agente:
 - ① Escolher agentes na sua vizinhança;
 - ② Imitar esses agentes;
 - ③ Actualizar a melhor experiência caso o valor actual seja melhor;
- ④ Iterar para 2 se ainda não atingimos o critério de paragem.

O PSO inicial

Parâmetros

Posição Posição actual \vec{x}_i ;

Velocidade Velocidade \vec{v}_i ;

Individualismo A melhor posição alcançada \vec{p}_i ;

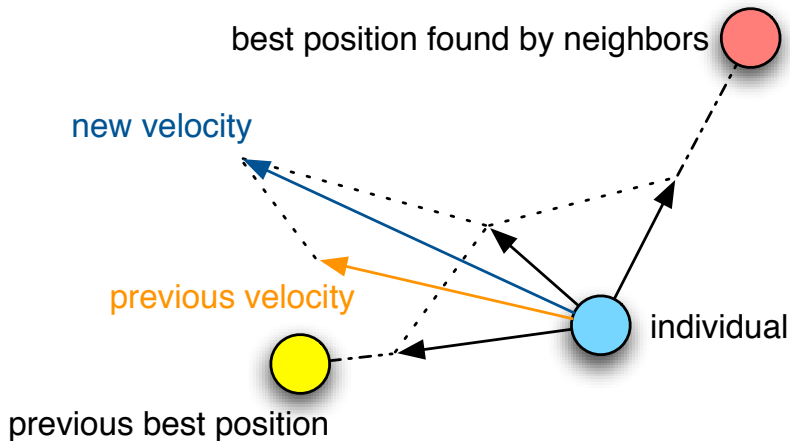
Conformismo A melhor posição encontrada pelos seus vizinhos \vec{p}_g ;

φ_1 and φ_2 Coeficientes de aceleração estocásticos of the stochastic acceleration terms.

Equações do PSO

$$\begin{cases} \vec{v}_i = \vec{v}_i + \vec{U}[0, \varphi_1](\vec{p}_i - \vec{x}_i) + \vec{U}[0, \varphi_2](\vec{p}_g - \vec{x}_i) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases}$$

Geração de Soluções no PSO



Velocidade Máxima

- A velocidade atinge facilmente valores muito elevados;
- Quando isso acontece, o algoritmo não estabiliza e não é capaz de encontrar boas soluções;
- Para contrariar esse efeito, introduziu-se o parâmetro V_{max} ;
- Esse parâmetro impede a velocidade de alcançar valores muito elevados;
- Tradicionalmente, ele é aplicado coordenada a coordenada;
- Se for demasiado elevado, os agentes tendem a não explorar boas soluções;
- Se for muito pequeno, os agentes andam com um passo demasiado pequeno e podem ficar encravados num óptimo local;
- Chegou-se à conclusão que se podia atribuir aos parâmetros φ_1 e φ_2 o valor de 2 para quase todas as aplicações e só necessitar de ajustar o valor atribuído a V_{max} ;
- Contudo, o valor atribuído a V_{max} era crucial para o bom desempenho do algoritmo.

Coeficiente Inercial

- Os investigadores ficaram rapidamente insatisfeitos com a arbitrariedade da atribuição do valor ao V_{max} ;
- Para além disso, o parâmetro influenciava negativamente a capacidade do PSO alternar entre exploração e aproveitamento;

PSO com Coeficiente Inercial

$$\begin{cases} \vec{v}_i = \alpha \vec{v}_i + \vec{U}[0, \varphi_1](\vec{p}_i - \vec{x}_i) + \vec{U}[0, \varphi_2](\vec{p}_g - \vec{x}_i) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases}$$

- Proposto por Shi e Eberhart;
- Normalmente utilizado decrescendo linearmente entre 0.9 e 0.4;
- Nestes casos, V_{max} é ajustado para os limites das coordenadas;
- Foram usados controladores difusos para ajustar dinamicamente o valor de α durante o processo de optimização;
- Outra opção consiste em atribuir valores aleatórios com média 0.5.

Coeficiente de Constrição

- Clerc efectuou uma análise do algoritmo estudando um sistema determinístico inspirado no PSO;
- Ele produziu um modelo generalizado do PSO e estudou as suas propriedades de convergência;
- Como resultado desse estudo ele propôs um modelo simplificado.

PSO com Coeficiente de Constrição

$$\begin{cases} \vec{v}_i = \chi(\vec{v}_i + \vec{U}[0, \varphi_1](\vec{p}_i - \vec{x}_i) + \vec{U}[0, \varphi_2](\vec{p}_g - \vec{x}_i)) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases}$$

onde

$$\chi = \frac{2k}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}$$

e $\varphi = \varphi_1 + \varphi_2$, $\varphi > 4$, $k \in [0, 1]$

Equivalência entre os Modelos Inercial e de Constrição

- Ambos os modelos são equivalentes;
- Os resultados de Clerc podem ser aplicados ao modelo inercial;
- Kennedy sugeriu a utilização do modelo de constrição atribuindo aos parâmetros $\varphi = 4.1$ ($\varphi_1 = \varphi_2 = 2.05$) e $k = 1$;
- Isso determina $\chi \approx 0.729$;
- Este modelo é equivalente a utilizar o modelo inercial com $\alpha \approx 0.729$ e $\varphi_1 = \varphi_2 \approx 1.49445$;
- Deixa de ser necessário atribuir um valor a V_{max} ;
- Mesmo assim há pessoas que continuam a atribuir a V_{max} o valor do limite das coordenadas.

PSO Canónico

Parâmetros

Posição Posição actual \vec{x}_i ;

Velocidade Velocidade \vec{v}_i ;

Individualismo A melhor posição alcançada \vec{p}_i ;

Conformismo A melhor posição encontrada pelos seus vizinhos \vec{p}_g ;

Parâmetros $\varphi_1 = \varphi_2 = 2.05, \chi = 0.729$.

PSO Canónico

$$\begin{cases} \vec{v}_i = \chi(\vec{v}_i + \vec{U}[0, \varphi_1](\vec{p}_i - \vec{x}_i) + \vec{U}[0, \varphi_2](\vec{p}_g - \vec{x}_i)) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases}$$

Exemplo de Execução

Parâmetros

População Normalmente 20, neste pequeno exemplo será de 3;

Função Objectivo $f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$;

Limites -5 a 5.

Exemplo de Execução

Parâmetros

População Normalmente 20, neste pequeno exemplo será de 3;

Função Objectivo $f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$;

Limites -5 a 5.

Inicialização

Posição Cada coordenada é escolhida segundo a distribuição uniforme entre os limites $[-5; 5]$;

Velocidade Nula;

Melhor Posição Igual à posição;

Aptidão Qualidade da solução inicial.

Exemplo de Execução

Parâmetros

População Normalmente 20, neste pequeno exemplo será de 3;

Função Objectivo $f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$;

Limites -5 a 5.

Exemplo de Inicialização

Posição 2.279641 4.246692;

Velocidade 0 0;

Melhor Posição 2.279641 4.246692;

Aptidão $(1 - 2.279641)^2 + 100(4.246692 - 2.279641^2)^2 = 91.90091$.

Exemplo de Execução

População Inicial

Núm	x_1	x_2	v_1	v_2	b_1	b_2	f
1	2.279641	4.246692	0	0	2.279641	4.246692	91.90091
2	2.130897	-3.140616	0	0	2.130897	-3.140616	5901.576
3	2.536688	-3.397041	0	0	2.536688	-3.397041	9668.845

Actualização do Agente 2

Posição $\langle 2.130897, -3.140616 \rangle$;

Velocidade $\langle 0, 0 \rangle$;

Melhor posição individual $\langle 2.130897, -3.140616 \rangle$;

Melhor posição do grupo $\langle 2.279641, 4.246692 \rangle$;

Exemplo de Execução

População Inicial

Núm	x_1	x_2	v_1	v_2	b_1	b_2	f
1	2.279641	4.246692	0	0	2.279641	4.246692	91.90091
2	2.130897	-3.140616	0	0	2.130897	-3.140616	5901.576
3	2.536688	-3.397041	0	0	2.536688	-3.397041	9668.845

Actualização do Agente 2

$$v_1 = 0.729 * (0 + 0.5809404 * 2.05 * (2.130897 - 2.130897) + 0.7322508 * 2.05 * (2.279641 - 2.130897)) = 0.1627724$$

$$v_2 = 0.729 * (0 + 0.8914667 * 2.05 * (-3.140616 + 3.140616) + 0.1816996 * 2.05 * (4.246692 + 3.140616)) = 2.005957$$

$$x_1 = 2.130897 + 0.1627724 = 2.293669$$

$$x_2 = -3.140616 + 2.005957 = -1.134659$$

$$f = (1 - 2.293669)^2 + 100 * (-1.134659 - 2.293669^2)^2 = 4092.013$$

Exemplo de Execução

População Inicial

Núm	x_1	x_2	v_1	v_2	b_1	b_2	f
1	2.279641	4.246692	0	0	2.279641	4.246692	91.90091
2	2.130897	-3.140616	0	0	2.130897	-3.140616	5901.576
3	2.536688	-3.397041	0	0	2.536688	-3.397041	9668.845

Actualização do Agente 2

- Como o valor da função objectivo (4092.013) é melhor do que o anterior (5901.576) actualizamos a melhor posição encontrada e a aptidão;

Exemplo de Execução

População Inicial

Núm	x_1	x_2	v_1	v_2	b_1	b_2	f
1	2.279641	4.246692	0	0	2.279641	4.246692	91.90091
2	2.130897	-3.140616	0	0	2.130897	-3.140616	5901.576
3	2.536688	-3.397041	0	0	2.536688	-3.397041	9668.845

Iteração 1

Núm	x_1	x_2	v_1	v_2	b_1	b_2	f
1	4.004912	7.495891	1.725271	3.249199	2.279641	4.246692	91.90091
2	2.293669	-1.134659	0.1627724	2.005957	2.293669	-1.134659	4092.013
3	4.774601	-4.119777	2.2379132	-0.7227357	2.536688	-3.397041	9668.845

Conceitos Importantes de PSO

Tipo Algoritmo para optimização real;

Imitação Cria novas soluções “deslocando-se” em direcção a soluções prévias;

Memória Guarda a melhor posição descoberta;

Elitismo Na verdade o que interessa para cada agente é a melhor posição que descobriu.

Passo adaptativo Consegue combinar alturas de exploração e aproveitamento.

Algoritmos Genéticos

- Existe uma população de indivíduos (soluções);
- Em cada iteração efectuam-se os seguintes passos:
 - ▶ Usa-se operadores de selecção (e.g. selecção baseada na roleta, torneio, ordenação de indivíduos);
 - ▶ Usam-se operadores de recombinação;
 - ▶ Usam-se operadores de mutação.
- Avalia-se a nova população;
- Cria-se a nova população.

Comparação com Algoritmos Genéticos

Semelhanças

- Ambos são algoritmos de otimização;
- Ambos podem ser utilizados desde que se consiga reduzir uma tarefa a um problema de otimização real;
- O esquema de funcionamento é semelhante;

Diferenças

- O PSO é menos versátil do que o GA;
- O PSO é normalmente mais eficiente em problemas de otimização real;
- Os operadores genéticos do GA substituem os indivíduos;
- O passo do PSO é naturalmente adaptativo;
- O GA tem vários operadores genéticos.

Procura Harmónica

- Há um armazém de harmonias (soluções);
- Para se gerar uma nova harmonia efectua-se os seguintes passos para cada dimensão:
 - ▶ Escolhe-se um valor aleatoriamente do armazém de harmonias e altera-se esse valor aleatoriamente ou;
 - ▶ Gera-se um valor dentro do intervalo válido.
- Avalia-se a nova harmonia;
- Esta nova harmonia substitui aquela que está no armazém com o pior resultado.

Comparação com Procura Harmónica

Semelhanças

- Ambos são algoritmos de optimização real;
- Ambos possuem conceitos elitistas (no caso da procura harmónica, a nova harmonia substitui a pior);

Diferenças

- Ao contrário do PSO, o passo da procura harmónica não é adaptativo;
- Cada coordenada é gerada aleatoriamente ou é uma alteração de uma das posições do armazém (não necessariamente do mesmo).

Evolução Diferencial

- Para cada indivíduo:

- ▶ Escolhem-se dois indivíduos aleatoriamente (distintos entre si e do indivíduo actual);
- ▶ Cria-se um vector que é composto pela soma do indivíduo actual com a uma proporção da diferença entre os dois vectores;
- ▶ Este vector entra numa operação de recombinação aleatória com o indivíduo actual;
- ▶ O vector resultante da operação de recombinação substitui o actual se a qualidade da solução for melhor.

Comparação com Evolução Diferencial

Semelhanças

- Ambos são algoritmos de otimização real;
- Ambos tem passo adaptativo;
- Alguns esquemas do DE são bastante parecidos com o PSO.

Diferenças

- O DE não tem velocidade;
- O DE tem um crossover probabilístico;
- O DE usa uma só equação e escolhe aleatoriamente vários agentes para determinar o passo da variação.

Utilização do PSO

Conceitos

Tarefa O problema a resolver tem que ser reescrito como um problema de optimização real;

Variáveis Cada solução do problema tem que ser representada como um dado conjunto de variáveis reais;

Restrições As restrições são normalmente tratadas na função objectivo atribuindo uma qualidade má aos casos que violem as restrições;

Função Objectivo Deve ser escolhida de forma a que numa dada vizinhança as soluções tenham qualidades semelhantes.

Exemplos

Treino de Redes Neurais

Variáveis Cada peso da rede é uma das variáveis do problema;

Restrições Não tem;

Função Objectivo O erro (e.g., percentagem de casos classificados correctamente num problema de classificação ou somatório dos quadrados dos erros num problema de regressão).

Segmentação usando K-means

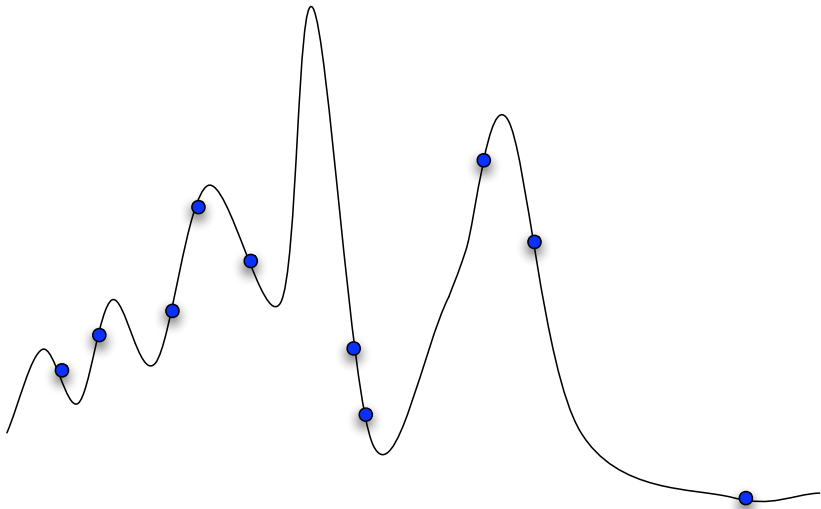
Tarefa Usar PSO para determinar as coordenadas iniciais dos clusters;

Variáveis Cada variável representa uma das dimensões de um dos clusters;

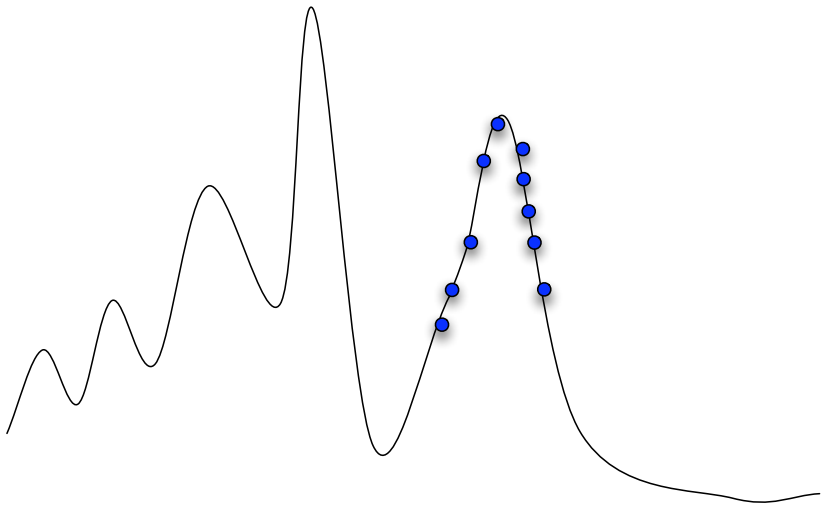
Restrições Não tem;

Função Objectivo Distância média que cada ponto ao centro do seu cluster.

Convergência Prematura



Convergência Prematura



Quais são as causas?

A otimização é um equilíbrio de duas forças:

exploração Ser capaz de encontrar áreas promissoras no espaço de procura;

aproveitamento Concentrar-se em encontrar melhores soluções nas áreas promissoras.

- Um algoritmo com demasiada exploração não é *eficiente*;
- Um algoritmo que se concentra demasiado no aproveitamento perde *diversidade* depressa;
- Se um algoritmo tem pouca diversidade, a procura pode *estagnar*.

Como resolver problemas de convergência prematura em PSO?

- O PSO tem tendência a convergir prematuramente;
- Pode-se resolver esse problema de duas maneiras:
- Aumentar a diversidade da população;
- Tentar perceber as causas dessa perda de diversidade.

Criticalidade Auto-Organizada

- Proposto por Løvbjerg e Krink;
- Quando dois agentes se aproximam demasiado, o seu valor crítico aumenta;
- O valor crítico de cada agente diminui ligeiramente a cada iteração;
- Quando o valor crítico de um agente atinge um determinado nível, ele explode, dispersando a sua criticalidade pelos agentes próximos e realoca-se no espaço de procura.

Criticalidade Auto-Organizada

- Proposto por Løvbjerg e Krink;
- Quando dois agentes se aproximam demasiado, o seu valor crítico aumenta;
- O valor crítico de cada agente diminui ligeiramente a cada iteração;
- Quando o valor crítico de um agente atinge um determinado nível, ele explode, dispersando a sua criticalidade pelos agentes próximos e realoca-se no espaço de procura.

Desvantagens

- Tenta aumentar a diversidade após ela ter desaparecido;
- Cria um parâmetro que é dependente do problema (a distância mínima);
- Tende a evitar que se encontrem soluções de qualidade elevada.

Enxâmes que Evitam Colisões

- Proposto por Blackwell e Bentley;
- Utilizado para seguir alvos móveis;
- Proposto por Krink, Vesterstrøm e Riget;
- Chamado partículas espacialmente extendidas;
- Os agentes fazem ricochete quando chocam.

Enxâmes que Evitam Colisões

- Proposto por Blackwell e Bentley;
- Utilizado para seguir alvos móveis;
- Proposto por Krink, Vesterstrøm e Riget;
- Chamado partículas espacialmente extendidas;
- Os agentes fazem ricochete quando chocam.

Desvantagens

- Tenta aumentar a diversidade após ela ter desaparecido;
- Cria um parâmetro que é dependente do problema (o diâmetro das partículas);
- Tende a evitar que se encontrem soluções de qualidade elevada.

Esticar Funções

- Proposto por Parsopoulos et al;
- Efectuar uma transformações de dois passos à função objectivo após detectar estagnação;
- Elevar a função para eliminar todos os mínimos locais abaixo do que acabou de ser encontrado;
- Alisar a área perto do mínimo local;
- Ambos os passos não alteram a função para as regiões acima do mínimo local;
- Depois da transformação reinicializar todas posições e velocidades de todos os agentes;
- This technique may introduce false local optima and misleading gradients

Esticar Funções

- Proposto por Parsopoulos et al;
- Efectuar uma transformações de dois passos à função objectivo após detectar estagnação;
- Elevar a função para eliminar todos os mínimos locais abaixo do que acabou de ser encontrado;
- Alisar a área perto do mínimo local;
- Ambos os passos não alteram a função para as regiões acima do mínimo local;
- Depois da transformação reinicializar todas posições e velocidades de todos os agentes;
- This technique may introduce false local optima and misleading gradients

Desvantagens

- Tenta a introduzir mínimos locais falsos e gradientes enganadores;
- Dependendo da forma como se detecta que se está num mínimo local pode evitar que se encontrem soluções de qualidade elevada.

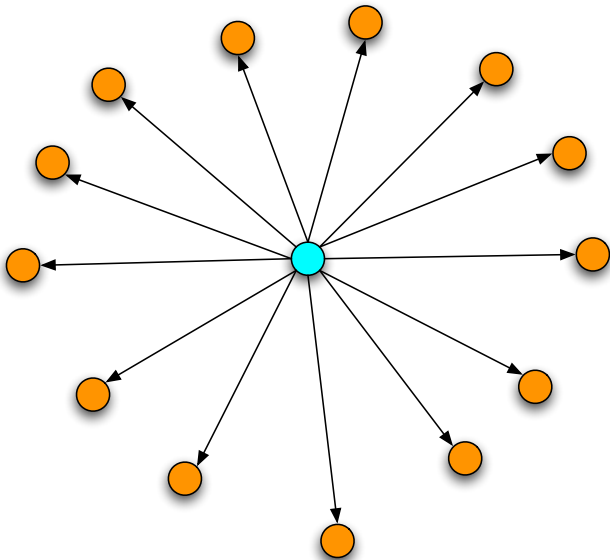
Porque há convergência prematura em PSO?

- O agente mais bem sucedido de *toda a população* é o responsável pela influência social;
- A boa solução descoberta atrai toda a população;
- Esta situação pode levar a convergência prematura.

Porquê Utilizar Vizinhanças?

- O algoritmo inicial não utilizava este conceito;
- A população usava o modelo *panmítico*;
- Todos os agentes se conhecem;
- Este modelo é conhecido em PSO por *Global Best* (sigla *gbest*)
- A convergência é rápida;
- A diversidade diminui depressa;
- Esta situação pode acarretar convergência prematura.

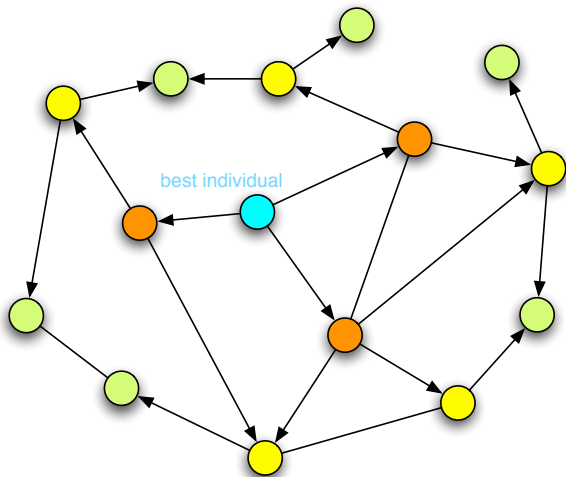
Exemplo de Influência Imediata



Conceito de Vizinhança

- Os agentes imitam os seus vizinhos mais bem sucedidos;
- Cada agente só influencia directamente os seus vizinhos;
- Os seus vizinhos só influenciarão os vizinhos deles quando se tornarem bem sucedidos;
- Esta situação favorece a segmentação: vizinhanças sociais diferentes podem explorar zonas distintas do espaço de procura;
- A vizinhança pode advir de:
 - Proximity Proximidade no espaço Cartesiano;
 - Social Partilha de laços sociais.

Exemplo do Fluxo de Influência



immediate
second degree
third degree

Vizinhança Cartesiana

- É necessário calcular a distância entre cada par de agentes em todas as iterações;
- A complexidade aumenta com o tamanho da população e o número de dimensões.

Vizinhança Social

- A estrutura da população é uma rede social;
- Representada como um grafo;
 - ▶ Os agentes são representados como vértices;
 - ▶ Os ramos ligam agentes que partilham um laço social.
- É mais eficiente do que a vizinhança Cartesiana.

Estatísticas de Grafos Importantes

Grau Número médio de vizinhos;

Distância Distância média entre dois agentes;

Sequência de Distribuição Número de vizinhos, de segundos vizinhos, de terceiros vizinhos, etc.

Características de uma Boa Topologia

Grau Relativamente pequeno;

Distância Nem muito pequena nem muito grande, está altamente correlacionada com o grau;

Sequência de Distribuição A maioria da população deve estar nos três primeiros níveis.

Exemplos de Topologias Razoáveis

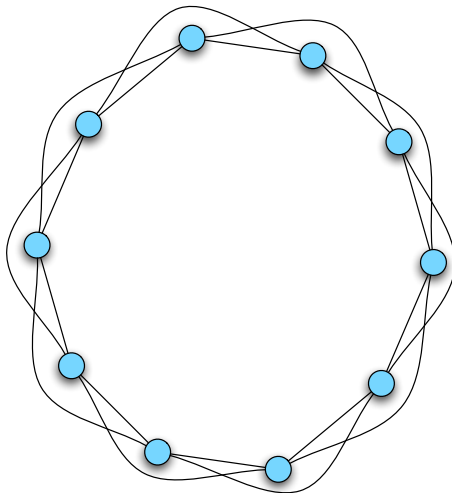


Figura : lbest2

Exemplos de Topologias Razoáveis

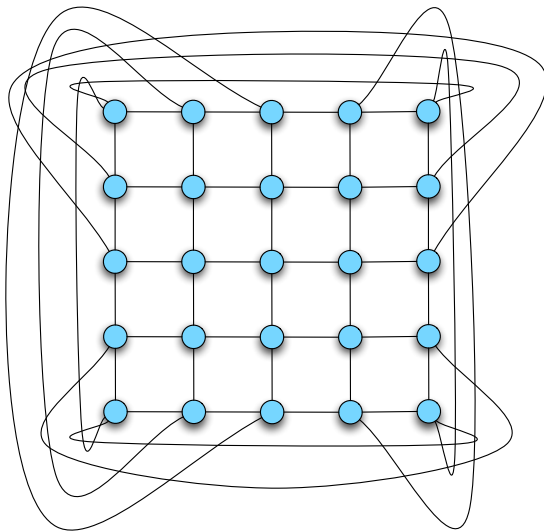


Figura : von Neumann

Topologias Dinâmicas

- Vizinhanças criadas automaticamente usando proximidade no espaço de procura (usando segmentação);
- Começar com um anel e ir juntando ligações até chegar a *gbest*;
- Hierarquias dinâmicas;
- Topologias que mudam aleatoriamente ao longo da execução;
- FDR-PSO: escolher uma partícula próxima que seja bem sucedida;

Imitação

- O algoritmo tradicional usa a contribuição do melhor indivíduo da vizinhança e a sua melhor experiência passada;
- A restante informação da vizinhança não é utilizada;
- A questão é se se perde informação importante ao desperdiciar essa informação.

- A polarização acontece quando os individuos adoptam os hábitos do grupo;
- As pessoas tendem a tornar-se mais parecidas com o grupo à medida que o tempo vai passando:
 - ▶ Vestem-se de forma semelhante;
 - ▶ Passam a comer as mesmas coisas;
 - ▶ Passam a gostar do mesmo tipo de música.

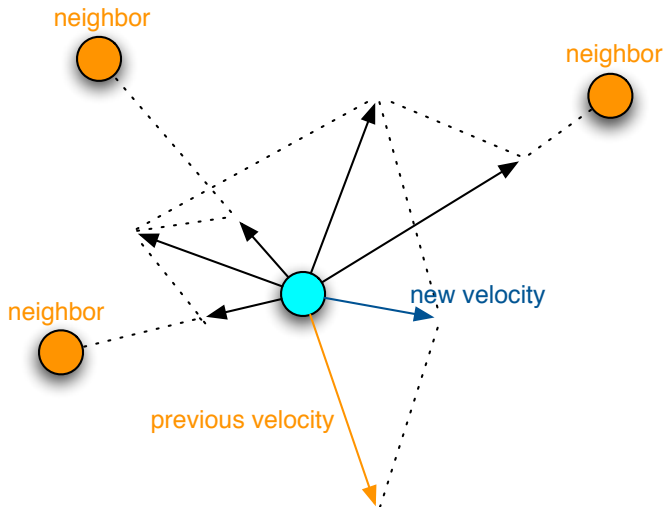
Fully Informed Particle Swarm

- Todas as contribuições da vizinhança são utilizadas;
- Os indivíduos imitam a normal social;
- A normal social é o centro de gravidade;
- \vec{p}_k é a melhor posição anterior do vizinho k ;
- \mathcal{V} é o conjunto de vizinhos.

Geração de novas posições do FIPS

$$\begin{cases} \vec{v}_{t+1} = \chi \left(\vec{v}_t + \frac{\sum_{k \in \mathcal{V}} \vec{U}[0, \varphi_{max}](\vec{p}_k - \vec{x}_t)}{|\mathcal{V}|} \right) \\ \vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \end{cases}$$

Geração de Soluções no FIPS



Pseudo-Algoritmo de Geração de Soluções do FIPS

- ➊ Para cada uma das coordenadas d ;
- ➋ soma_pos = 0, soma_coef = 0;
- ➌ Para cada vizinho do agente $v_k \in \mathcal{V}$;
 - ➏ Gerar um valor distribuido aleatoriamente entre 0 e 1 $c_k = U[0, 1]$;
 - ➐ soma_pos = soma_pos + $c_k * p_k$;
 - ➑ soma_coef = soma_coef + c_k ;
- ➍ pos_med = soma_pos/soma_coef;
- ➎ var_phi = $\frac{\varphi \text{ soma_coef}}{|\mathcal{V}|}$;
- ➏ $v_d = \chi * (v_d + \text{var_phi} * (\text{pos_med} - x_d))$;
- ➐ $x_d = x_d + v_d$;

Diferenças no FIPS

- Não existe contribuição individual;
- Todos os agentes contribuem;
- O número de vizinhos é muito importante:
 - ▶ Se forem demasiados, o FIPS não funciona;
 - ▶ O número correcto de vizinhos deve ser entre 3 e 4.
- Com certas topologias de população, o FIPS consegue resultados superiores ao PSO canónico.

O Dogma Central de PSO

Kennedy sugeriu que o paradigma pode ser escrito da seguinte forma:

$$\text{NEW POSITION} = \text{CURRENT POSITION} + \text{PERSISTENCE} + \text{SOCIAL CENTRAL TENDENCY} + \text{SOCIAL DISPERSION}$$

NEW POSITION A nova posição do agente;

CURRENT POSITION A posição actual;

PERSISTENCE A tendência em seguir na mesma direcção;

SOCIAL CENTRAL TENDENCY Informação sobre a tendência do grupo; pode ser por exemplo a melhor contribuição do grupo ou a média das contribuições;

SOCIAL DISPERSION Uma medida da capacidade de exploração do algoritmo; esta medida depende normalmente de quão agrupados estão os agentes na vizinhança.

O Dogma Central de PSO

Kennedy sugeriu que o paradigma pode ser escrito da seguinte forma:

$$\text{NEW POSITION} = \text{CURRENT POSITION} + \text{PERSISTENCE} + \text{SOCIAL CENTRAL TENDENCY} + \text{SOCIAL DISPERSION}$$

- Esta formula facilita a compreensão do que pode ser considerado PSO;
- Ela abre caminho para novas experiências;
- O conceito de persistência permite tentar outras coisas como PSOs sem velocidade;
- A tendência social central pode ser escrita de muitas maneiras; pode-se utilizar só a contribuição do melhor agente na vizinhança ou de todos como no FIPS;
- A dispersão social indica outras formas aos investigadores de criar algoritmos usando geradores aleatórios com distribuições estatísticas: e.g. triangular, Gaussiana, Cauchy.

Bare-Bones PSO

A Experiência

- Kennedy fez uma experiência com o PSO canónico em que manteve p_i e p_g constantes;
- A forma da curva é uma curva de sino onde as pontas são mais grossas do que uma curva de Gauss;
- Inspirado nessa experiência, Kennedy criou o Bare-Bones PSO.

Bare-Bones PSO

$$x_{t+1} = N\left(\frac{p_i + p_g}{2}, |p_i - p_g|\right)$$

Gaussian FIPS

Resultados do Bare-Bones

- A experiência com o Bare-Bones foi interessante mas os resultados não eram competitivos;
- A interpretação desses resultados e o facto da cauda das distribuições do PSO ser maior, sugere algum tipo de persistência;
- Tal como no FIPS a influência social é a média das contribuições dos vizinhos.

Gaussian FIPS

$$p_m = \frac{\sum_{k \in \mathcal{N}} p_k}{|\mathcal{N}|}$$
$$x_{t+1} = \chi(x_t - x_{t-1}) + N\left(p_m, \frac{|p_m - p_i|}{2}\right)$$

FIPS Gaussiano com Baricentro Estocástico

Motivação

- Uma evolução do Gaussian FIPS;
- Incorpora mais uma ideia do FIPS: o baricentro estocástico;
- O baricentro estocástico é uma média pesada com coeficientes aleatórios das contribuições dos vizinhos.

FIPS Gaussiano com Baricentro Estocástico

$$\begin{aligned}c_k &= U[0, 1], k \in \mathcal{N} \\ p_m &= \frac{\sum_{k \in \mathcal{N}} c_k p_k}{\sum_{k \in \mathcal{N}} c_k} \\ x_{t+1} &= \chi(x_t - x_{t-1}) + N\left(p_m, \frac{|p_m - p_i|}{2}\right)\end{aligned}$$

Problemas de Optimiza  o Din mica

- Os problemas de optimiza  o podem ser din micos;
- O problema muda com o tempo (e.g., uma solu  o de uma rota de um taxi pode mudar por mudarem as condi  es de transito);
- O objectivo num problema de optimiza  o din mico   seguir o  ptimo o m ximo poss vel adaptando-se  s mudan as;
- A maioria das t cnicas utilizadas nestes problemas pode ser utilizada nos casos em que queiramos descobrir todos os  ptimos locais de um problema de optimiza  o.

Vantagens do PSO

- Como uma população de soluções, o algoritmo pode manter informação útil sobre as características do ambiente;
- Como o PSO tem uma velocidade de convergência elevada, deveria conseguir adaptar-se rapidamente a uma alteração no ambiente;
- Algumas experiências iniciais mostraram que o PSO conseguia localizar e seguir alvos móveis.

Conceitos Base

- Detectar mudanças no ambiente e reavaliar os agentes caso isso aconteça;
- Aumentar a diversidade para ser capaz de descobrir novos picos quando o ambiente mudar;
- Manter vários enxâmes em simultâneo (de preferência devem existir tantos enxâmes como picos);
- Impedir que vários enxâmes convirjam para o mesmo pico.

Implementação dos Conceitos

- 1 Manter vários enxâmes em simultâneo (de preferência devem existir tantos enxâmes como picos);
- 2 Periódicamente a melhor posição de cada agente é reavaliada para verificar se o valor da função objectivo mudou;
- 3 Manter agentes PSO normais e agentes quânticos (que gravitam probabilisticamente à volta dos primeiros) para aumentar a diversidade;
- 4 Alternativamente usar agentes com cargas eléctricas de forma a repelirem-se quando estão muito próximos;
- 5 Calcular a distância entre dois enxâmes (distância entre os melhores agentes de ambos os enxâmes) e caso esta seja menor do que um dado parâmetro reinicializar o enxâme pior;
- 6 Calcular o diâmetro de cada enxâme; se todos os enxâmes convergirem (o diâmetro for menor do que um dado valor) reinicia-se o pior enxâme.

Fim

Questões?