



INF 1771 – Inteligência Artificial

Trabalho 01 - Dicas

2016.2



Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>



APIs para Interfaces Gráficas

- C++
 - SDL+
 - Playlib (iCAD)
 - Cocos2d-x
- JAVA
 - Swing
 - LWJGL
- C#
 - System.Drawing
 - Monogame.net (continuação do antigo Microsoft XNA)
 - Cocos2d-XNA
- Python
 - Gasp
 - PyGame
 - Coco2d(python)

Pseudocódigo – Simulated Annealing

Função Simulated-Annealing(*Problema*, *Escalonamento*) **retorna** um estado que é o máximo local

Início

EstadoAtual \leftarrow Criar-Nó(*Problema*[EstadoInicial])

loop do

$t = t + 1$

$T \leftarrow$ *Escalonamento*[t]

Se $T = 0$ **então retorna** EstadoAtual

 Próximo \leftarrow seleciona um sucessor do EstadoAtual aleatoriamente

$\Delta E \leftarrow$ Próximo[Valor] - EstadoAtual[Valor]

se $\Delta E > 0$ **então** EstadoAtual \leftarrow Próximo

senão EstadoAtual \leftarrow Próximo somente com probabilidade $e^{\Delta E/T}$

Fim

- Combina a subida de encosta com um **percurso aleatório** resultando em eficiência e completeza.
- Subida de encosta dando uma “chacoalhada” nos estados sucessores;
- Idéia: **Escapar do máximo local** aceitando alguns movimentos “ruins” mas gradualmente diminuindo sua frequência usando o critério de aceitação de **Boltzmann** $e^{(\Delta E/T)}$

Pseudocódigo – Simulated Annealing

```
Solution Si = Disturb(S);  
double fSi = Si.Calculate();  
double fS = S.Calculate();  
  
double dFi = (fSi - fS);  
  
double exp = Math.Exp(-Math.Abs(dFi) / (double)T);  
double rnd = Randomize();  
  
if ((dFi < 0) || (exp > rnd))
```

Pseudocódigo – Algoritmo Genético

Função *GA(PopulacaoInicial)* **retorna** um estado que é o máximo local

Início

PopulacaoAtual = PopulacaoInicial

Faça

Faça

 individuoPai = Roleta (populacaoAtual)

 individuoMae = Roleta (populacaoAtual)

 individuoFilho = CrossOver(individuoPai, individuoMae)

 probMutacao = seleciona um valor aleatoriamente [0,1]

Se probMutacao <= 0.05 **Então**

 individuoFilho = Mutacao(individuoFilho)

 novaPopulacao += individuoFilho

 individuos = individuos + 1

Enquanto (individuos < MaxIndividuos)

 populacaoAtual = novaPopulacao

 geracao = geracao + 1

Enquanto (Geracao < MaxGeracoes)

Fim