

Computação Bioinspirada

AULA 1 – APRESENTAÇÃO DA DISCIPLINA

PROF. PAULO SALGADO

Aula de hoje

Falar sobre

- Objetivos
- Ementa
- Avaliação
- Roteiro Proposto (Cronograma)
- Referências
- Periódicos da Área

Objetivos

Mostrar aos alunos um novo paradigma da computação.

- Computação Evolutiva (Algoritmos Evolucionários)
- Inteligência de Enxames (Algoritmos de Enxame)

Dar subsídios para os alunos para o desenvolvimento de sistemas computacionais adaptativos.

Ampliar o horizonte de conhecimentos a respeito de técnicas e algoritmos de ponta utilizados em pesquisas atuais.

Ementa

Visão global da Computação Bioinspirada, em particular com a introdução dos conceitos básicos e avançados no ramo da

- Computação Evolutiva (Evolucionária).
 - Algoritmos Genéticos
 - Estratégias Evolutivas
 - Programação Genética
- Inteligência de Enxames
 - PSO
 - ACO
 - Firefly Algorithm
 - ...
- Algoritmos Meméticos

Avaliação

Projetos:

Dois projetos regulares de implementações com apresentação de seminários;

Um mini-projeto na 1ª unidade

Cada um destes projetos podem ser realizados individualmente ou em equipe de no máximo 3 integrantes;

Roteiro Proposto

1. Introdução

- a. Breve Histórico
- b. Inspiração Biológica
- c. Por que Computação Bioinspirada?

2. O que é um algoritmo evolutivo?

- a. Algoritmo Evolutivo (AE)
- b. Componentes de um AE
- c. Exemplos e aplicações

3. Algoritmos Genéticos (AG)

- a. Funcionamento dos AGs
- b. Representação de indivíduos
- c. Operadores Genéticos
- d. Modelos de População

Realização do Mini-Projeto

4. Estratégias Evolutivas (EE)

- a. Representações
- b. Processos de Mutação
- c. Processos de Recombinação
- d. Auto-Adaptação

Roteiro Proposto

5. Programação Genética

Realização do Primeiro Projeto

6. Algoritmos Meméticos

7. Algoritmos de Enxames

- a. *Particle Swarm Optimization (PSO)*
- b. *Ant Colony Optimzation (ACO)*
- c. *Firefly Algorithm*

Realização do Segundo Projeto e Seminários

Referências

EIBEN, A. E.; SMITH, J. E. Introduction to Evolutionary Computing. Springer, 2003.

BARTZ-BEIELSTEIN, T. Experimental Research in Evolutionary Computation: The New Experimentalism. Springer, 2006

KALLEL, L.; NAUDTS, B.; ROGERS, A. Theoretical Aspects of Evolutionary Computing. Springer, 2001.

GROSAN, C.; ABRAHAN, A.; ISHIBUCHI, H. Hybrid Evolutionary Algorithms. Springer, 2007.

BONABEAU, E.; DORIGO, M.; THERAULAZ, G. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999.

CASTRO, L. N.; Computação Natural: Uma Jornada Ilustrada. Física, 2010.

Periódicos da Área

Periódicos de grande repercussão

- **IEEE Transactions on Evolutionary Computation**
- **Evolutionary Computing (M.I.T. Press)**

Outros periódicos

- <http://www.macs.hw.ac.uk/~ml355/journals.htm>

Introdução à Computação Bioinspirada

Computação clássica vs computação biológica

Motivação para a computação biologicamente inspirado

Visão geral de algumas **coisas** de inspiração biológica

O que veremos na aula de hoje

O que é “computação clássica”? Para que tarefas essa “computação” é adequada?

Onde a “computação clássica” é deficiente?

Como a computação e a resolução de problemas se inspirar em sistemas biológicos.

Valorização do fato de que muitos exemplos de cálculos realizados por sistemas biológicos ainda não são correspondidos por o que podemos fazer com os computadores.

Por que desenvolver computação baseada na natureza?

Visão geral dos principais métodos utilizados com sucesso na CB

Computação Clássica x Computação Bioinspirada

Como você dizer diferenciar um cão e um gato?

Como poderíamos projetar um software para um robô que poderia fazer uma xícara de chá em sua cozinha?

Computação Clássica é boa quando:

- Envolve processamento de números
- É necessário ter suporte a pensamento (de papel e caneta glorificado)
- Raciocínio baseado em regras
- Há a repetição constante de ações bem definidas.

Computação Clássica é **ruim** em:

- Reconhecimento de padrões
- Robustez a danos
- Lidar com informações vagas e incompletas
- Adaptação e melhoria com base na experiência

Por que não podemos ter um software que pode resolver os seguintes problemas bem?

Localizar automaticamente uma pequena confusão na arquibancada em um campo de futebol

Classificar uma espécie de planta a partir de uma fotografia de uma folha.

Fazer uma xícara de chá?

Reconhecimento de Padrões e Otimização

Essas duas tarefas tendem a subir muito quando pensamos no que gostaríamos de ser capaz de fazer com o software, mas geralmente não podem fazer.

Reconhecimento de Padrões e Otimização são tarefas bem executadas na Biologia (seres vivos).

Então, parece que é uma boa ideia estudar como essas tarefas são resolvidas pela Biologia - ou seja, (geralmente) como o cálculo é feito por “máquinas biológicas”

Conceitos básicas sobre Reconhecimento de Padrões e Otimização

Reconhecimento de Padrões é frequentemente chamado de

- Classificação
- Agrupamento

Formalmente um problema de classificação é enunciado:

Tem-se um conjunto de “objetos”: S (imagem, vídeos, cheiros, vetores,)
Temos n possíveis classes: c_1, c_2, \dots, c_n , e sabemos que qualquer “objeto” em S deve ser classificado (rotulado) com precisão entre uma dessas classes

Em termos computacionais, a tarefa é definida como:

- Nós podemos projetar um processo computacional que leva um “objeto” s (a partir de S) como uma entrada, e sempre envia o rótulo da classe correta para s ?