

Programação genética cartesiana

Frederico Möller

April 15, 2021

1 Complexidade de problemas

- Problemas NP-completos
- Métodos Heurísticos

2 Computação evolutiva

- Conceito de programação evolutiva
- Algoritmo Genético
- Programação genética
- Aplicações na engenharia

3 Programação Genética Cartesiana

- O que é a CGP?
- CGP aplicada à projetos de circuitos lógicos
- Problemas da CGP
- Melhorias na CGP

Problemas NP-completos

- Complexidade aumenta de forma não polinomial
- Grande aumento nos recursos computacionais resultam em muito aumento no desempenho de sua resolução
- A maioria dos problemas das STEM são desse tipo.

- Sacrificio da melhor resposta
- Boa resposta em tempo hábil
- Economia de recursos

O que é programação evolutiva?

- Campo da ciência da computação
- Usa de artificios inspirados na teoria da evolução
- Algoritmos genéticos
- Programação genética

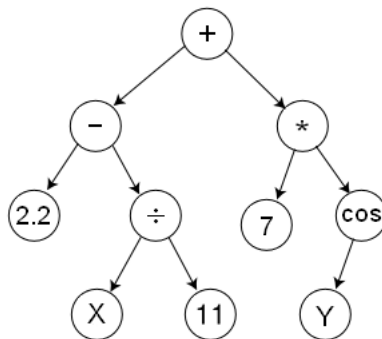
- Características do individuo mapeadas em genes
- Os atributos evoluem
- Representação de vetor

Evolução do conceito de inteligência



Figure: De: Wikipedia

- Evolução de programas
- Genes representam estruturas de manipulação de informação
- Representação em grafos e árvores



$$\left(2.2 - \left(\frac{X}{11} \right) \right) + \left(7 * \cos(Y) \right)$$

Figure: De: Wikipedia

- Projeto de antenas, colunas, asas de aeronaves, circuitos analógicos e digitais.
- Projeto de controladores.
- Competidor forte do aprendizado por reforço (RL) para caminhada de robôs bípedes.
- Otimização de projetos em relação à robustês.
- Projeto de adaptação de sistemas complexos para ambientes hostis.
- Ótimização de problemas multiobjetivos.

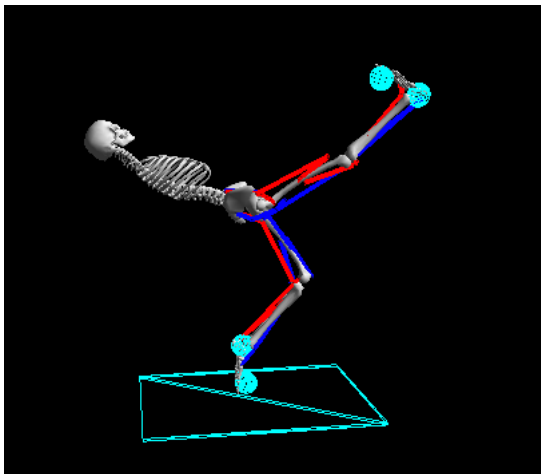
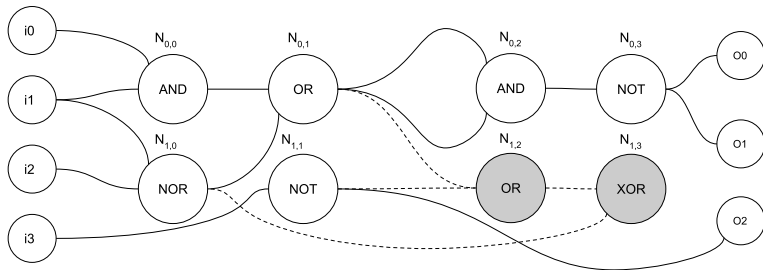


Figure: De Towards data science

O que é a CGP?

- Método de programação genética
- Nós distribuídos em linhas e colunas
- Nós com endereços de entradas e funções lógicas
- CGP clássica: nó pode se ligar apenas em nós de colunas anteriores

O que é a CGP?



CGP aplicada à projetos de circuitos lógicos

- Compete com o ESPRESSO
- Demanda mais recursos
- Apresenta melhores resultados
- Modular
- Espaço para melhorias

CGP aplicada à projetos de circuitos lógicos

- Mu pais e lambda individuos
- Padrão $\mu = 1$ $\lambda = 4$
- Cada pai gera lambda filhos
- Pai permanece até gerar filho melhor ou igual

- Problemas com crossover
- Indivíduos gerados por mutação
- PM vs SAM

- Custosa em tempo com relação ao Espresso
- Dificuldade de gerar circuitos factíveis em alguns problemas
- Verificação por tabela verdade: muito custosa
- Mutação não enviesada como unica forma de gerar novos individuos

- Uso de BDDs
- Paralelização
- Propostas de Crossover
- Enviesamento de mutações

fredericomollerper@gmail.com