

IA013 – Projeto Computacional 2

Computação Bio-Inspirada em Otimização

Contínua e Discreta

Peso da Lista: 10 || Data de entrega: 06/11/2008

1 Introdução

Embora possam ser utilizados programas prontos, dada a simplicidade da implementação de códigos básicos, o objetivo aqui é que o aluno desenvolva os seus próprios programas computacionais.

O relatório é individual e deve ser apresentado na forma de um artigo técnico-científico, o qual deve incluir:

- título;
- autor + e-mail + filiação;
- resumo;
- introdução (motivação + finalidade + oportunidade);
- seções técnicas e conceituais (posicionamento da pesquisa, base conceitual, metodologia, resultados, análise);
- conclusão + perspectivas futuras;
- agradecimentos;
- referências bibliográficas.

Não está sendo indicado nenhum estilo de edição a ser seguido. Quanto ao número de páginas, espera-se algo entre 12 e 20 páginas (quando em 2 colunas, letras no tamanho 10pt e espaçamento simples entre as linhas), incluindo figuras, tabelas e pseudo-códigos. Para outras configurações de edição, favor calcular aproximadamente o número de páginas equivalente à especificação acima.

Quando houver código e dados associados ao projeto, estes devem ser entregues em versão eletrônica.

2 Plano de Trabalho

Desenvolva uma versão ACO min-max para o problema do caixeiro viajante.

Desenvolva um algoritmo genético para o problema do caixeiro viajante, incluindo múltiplos operadores de mutação e crossover, além de propostas para busca local.

Compare as versões ACO min-max e GA+busca local com o Concorde, para três níveis de instâncias: fácil, mediana, difícil. Lembre-se que a dificuldade não está necessariamente vinculada ao número de cidades, mas à disposição das cidades no plano.

Escolha um outro tipo de problema de otimização combinatória e opte por algum dos algoritmos discutidos em aula para abordá-lo. Exemplo: problema de bipartição de números empregando simulated annealing.

Implemente uma versão do algoritmo PSO e escolha um problema de otimização em espaços contínuos para verificar sua performance. Exemplo: ajuste de pesos de redes neurais artificiais. Para efeito de visualização, pode ser a função abaixo, incluindo as curvas de nível.

Implemente o algoritmo opt-aiNet e aplique na solução do problema de maximização da função: $f(x, y) = x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) + 1$, no intervalo $[-1, +1] \times [-1, +1]$.

Implemente os algoritmos aiNet e ARIA para agrupamento de dados. Aplique a problemas em duas dimensões, com regiões distintas de densidade de dados. Pegue o tamanho da população final resultante e inicialize o k do algoritmo k -means. Compare os resultados. Não é necessário implementar minimum spanning tree (MST).

Sugestão: Para problemas combinatórios (espaços de otimização discretos), escolher conjuntos de dados dentre os disponíveis no repositório TSPLIB (ou similares):

<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

3 Pesquisadores de apoio

Além do professor da disciplina, os alunos da disciplina podem solicitar apoio técnico e tirar dúvidas com os seguintes pesquisadores doutorandos da FEEC/Unicamp:

- Wilfredo J. Puma-Villanueva (wilfredo@dca.fee.unicamp.br)
- Pablo A. D. de Castro (pablo@dca.fee.unicamp.br)
- Guilherme P. Coelho (guilherme.coelho@gmail.com)