

Prova - Computação Evolucionária

Henrique Queiroz do Amaral - 15.2.8014

03 de Julho de 2018

1 Introdução

A intenção deste documento é apresentar o desenvolvimento da atividade proposta da Prova da disciplina de Computação Evolucionária.

O modelo de algoritmo-base utilizado foi o Algoritmo Genético implementado em sala de aula. Os dois atributos utilizados para hibridização envolvem técnicas dos algoritmos Memético e Evolução Diferencial

2 Desenvolvimento

O primeiro passo consistiu em criar um novo problema (Schwefel) e definir alguns dos parâmetros, tal como os limites inferior e superior. Em seguida, alterando a estrutura do AG implementado em sala de aula [2], foi realizada a primeira hibridização baseada na ideia proposta no material da UFPE [1], de otimizar a inicialização da população do Algoritmo Genético através da seleção em populações aleatórias pré-fabricadas. Segundo o material, esse tipo de inicialização pode "direcionar a busca para regiões promissoras do espaço de busca, levando a aumentar a efetividade", sendo que de fato, houve uma melhora discreta na aproximação do resultado ótimo após a implementação da estratégia.

A segunda ideia de hibridização aborda os padrões vistos em sala de aula e consiste em acrescentar ao AG, parte das operações do Differential Evolution (DE). Foi transportada a ideia do indivíduo Trial, aplicando em seguida às operações entre Trial e Target, a ideia do cruzamento genético, característica do AG.

2.1 Inicialização Seletiva de População

Para a implementação da técnica de Inicialização Seletiva da População, foram adicionados métodos à classe População. O método

relevante é o `criarMemético()`. As instruções deste método basicamente geram duas populações pseudo-aleatórias, "embaralham" os indivíduos de forma a não serem distinguidos por ordem, onde seleciona e remove o primeiro de cada população. Em seguida faz a comparação nomeada ilustrativamente de cálculo de Fitness, onde verifica-se qual a porcentagem média de cada cromossomo do indivíduo está em relação ao valor ideal do problema para cada variável (420,9687). O indivíduo melhor colocado é então adicionado à população final, que será trabalhada pelo algoritmo.

2.2 Operações do DE implementadas no AG

A hibridização entre AG e DE deu-se pelo fato de haverem recursos entre ambos que podem ser combinados sem que se perca a característica original do algoritmo. Neste caso, a aplicação das operações sobre o indivíduo Trial, natural do algoritmo DE, no contexto do AG não afetam suas características de transmissão genética de informação (chance de crossover entre progenitor 1 e 2) e probabilidade de mutação ao gerar um novo indivíduo.

Para tentar preservar essas características de ambos algoritmos, foram adicionados os procedimentos com o trial (seleção, perturbação, mutação e crossover com o target) e em seguida, foi realizado o tratamento natural de possibilidade de crossover e mutação do AG, porém considerando o `Individuo1` como o Trial.

3 Considerações Finais

A aplicação dos métodos de hibridização ao Algoritmo Genético não se mostraram muito efetivas na solução do problema, com leve aproximação ao valor ótimo, como pode ser observado na tabela de resultados entregue separadamente.

4 Referências

- [1] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing. Capítulo 10, Acesso em: <http://cin.ufpe.br/psgmn/Bioinspirada/CE-10-ae-hibridos.pdf> às 21:00 do dia 29 de junho de 2018
- [2] Github - Fernando B. Oliveira - CSI557-Computacao-Evolucionaria. Acesso em: <https://github.com/fboliveira/CSI557-Computacao-Evolucionaria/tree/master/Codes/ECModel>