

UNIVERSIDADE FEDERAL DE OURO PRETO
DEPARTAMENTO DE CIÊNCIAS EXATAS E APLICADAS

Prova 1 - Implementação de uma versão híbrida de um Algoritmo Evolutivo

Otávio de Castro - 14.1.8151
Professor: Fernando Bernardes de Oliveira
Disciplina: Computação Evolucionária

Fevereiro/2018

1 Introdução

A proposta desta prova foi a implementação de uma versão híbrida de um algoritmo evolutivo, e para sua validação foi utilizado a função de avaliação conhecida por "Schwefel's function 7".

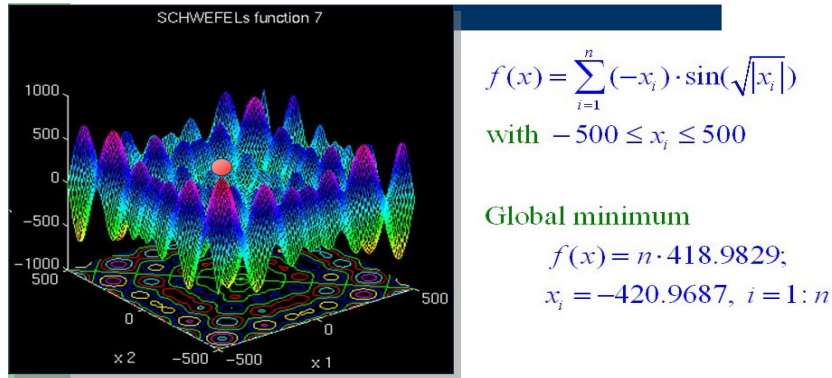


Figure 1: Schwefel's function 7

2 Algoritmo Proposto

O algoritmo híbrido desenvolvido tem com base principal o método da Estratégia Evolutiva, no qual foram acrescentadas operações contidas nas versões canônicas do Algoritmo Genético (crossover) e da Evolução Diferencial (perturbação + mutação), como visto no pseudocódigo abaixo:

- Gera pop inicial aleatoriamente e avalia
 - De $g = 1$ até o critério de parada, faz
 - Para cada pai, gerar λ/μ filhos
 - Se $\text{rnd} < \text{taxa de mutação}$
 - indv filho = clone do pai
 - mutaçãoES no filho
 - avalia FO do filho mutado e add na novapop
 - Se $\text{rnd} < \text{taxa de processo híbrido}$

- cria indivíduo trial
 - analisa a perturbação entre pai e filho mutado e insere em trial
 - mutaçãoDE (trial = indv aleatorio + 0.05 * fator de mutaçãoDE)
 - gera filho2 por crossover1pt (pai,trial)
 - avalia FO do filho2 e add na novapop
 - Adiciona novapop em pop
 - Ordena pop e realiza o corte
- Atualiza melhor solução

Figure 2: Pseudocódigo do algoritmo híbrido

3 Casos de Teste e Resultados

Os primeiros testes foram feitos através de uma comparação do algoritmo de Estratégia Evolutiva "puro" (caso1) e sua versão híbrida (caso2). Os seguintes parâmetros foram utilizados para ambos (os fatores pMutacaoDE, pH e Crossover não estão incluídos no caso1):

- Variáveis $n = 40$;
- Tamanho da população = 100 indivíduos;
- Gerações = 300;
- Lambda = 1000;
- pMutacaoES = 0.1;
- pMutacaoDE = 0.1;
- pH = 0.3;
- Crossover1pt = 0.1; (é sempre = pMutacaoES)

Os seguintes resultados foram obtidos:

- Melhor solução = C1: -13139.4 — C2: -14861.34
- Pior solução = C1: -9064.691 — C2: -12370.87
- Média resultados = C1: -10896.53 — C2: -13435.27
- Média tempo = C1: 1427ms — C2: 1962.667ms

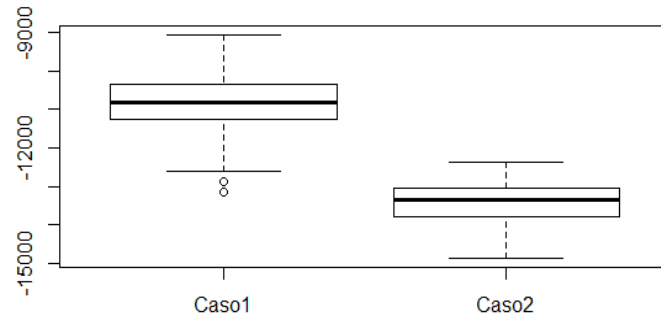


Figure 3: Boxplot Resultado ES x Híbrido

Teste T:

- Resultado: $p\text{-value} \leq 2.2e-16$ — $p\text{-valueL} = 1$ — $p\text{-valueG} \leq 2.2e-16$
- Tempo: $p\text{-value} \leq 2.2e-16$ — $p\text{-valueL} \leq 2.2e-16$ — $p\text{-valueG} = 1$

Através da análise visual pelo BoxPlot gerado, e pelos valores p obtidos no teste T, conclui-se que o algoritmo híbrido se mostrou significamente mais eficaz apesar de ser um pouco mais lento, devido as operações que foram adicionadas ao mesmo.

Outros testes "extras" foram feitos apenas no algoritmo híbrido realizando-se uma variação dos parâmetros:

- Variáveis $n = 40$;
- Tamanho da população = 100 indivíduos;
- Gerações = 300;
- Lambda = C1: 1500 — C2: 1000;
- $p\text{MutacaoES} = 0.1$;
- $p\text{MutacaoDE} = \text{C1: } 0.05 \text{ — C2: } 0.2$;
- $pH = \text{C1: } 0.5 \text{ — C2: } 0.1$;
- $\text{Crossover1pt} = 0.1$; (é sempre = $p\text{MutacaoES}$)

Os seguintes resultados foram obtidos:

- Melhor solução = C1: -14934.03 — C2: -15679.8
- Pior solução = C1: -12490.67 — C2: -12306.34
- Média resultados = C1: -13958.7 — C2: -14120.52
- Desvio padrão = C1: 590.8373 — C2: 659.776
- Média tempo = C1: 4455.667ms — C2: 3838.267ms

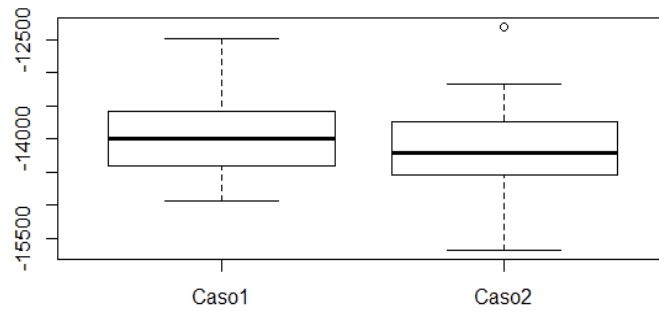


Figure 4: Boxplot Resultado Híbrido C1 x C2

Teste T:

- Resultado: p-value = 0.3211 — p-valueL = 0.8394 — p-valueG = 0.1606
- Tempo: p-value = 0.0001068 — p-valueL = 0.9999 — p-valueG = 5.338e-05

Através destes testes concluímos que os dois casos não tiveram nenhuma diferença considerável, com exceção do tempo, onde o caso 2 foi superior, o que pode ser explicado pelo fato da utilização de um *lambda* maior no primeiro.